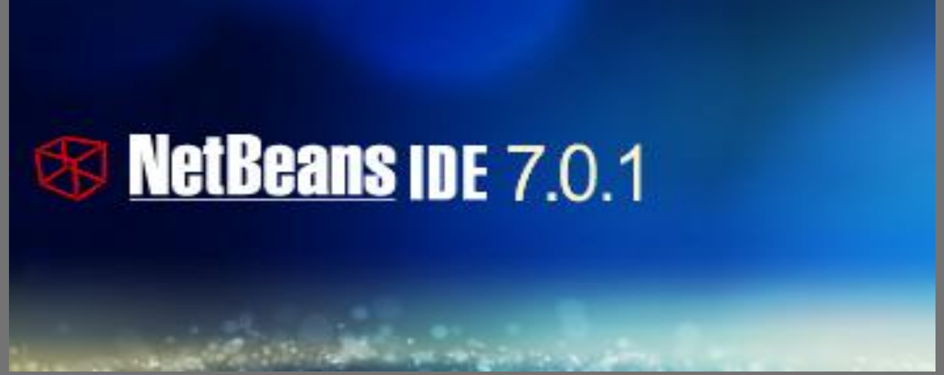
2014

تعلم الجافا الرسومية بإستخدام



بواسطة : أحمد مناع

هذا الكتاب سيأخذ بيدك الى بداية أحتراف لغة الجافا بخطوات بسيطة و سهلة الفهم بعيدا عن التعقيد الاكاديمى ....يبدأ معك من بداية تجهيز حاسبك بمنصة الجافا ثم يبد فى سرد الاساسيات بطريقة سهلة و مختصرة فى الوقت ذاته ثم نبدا فى التعرف على استخدام المكونات الرسومية باستخدام السحب و الاسقاط وكيفية التعامل مع أهم المكونات الرسومية و أكثرها شيوعا و إستخداما ...فى باب أخر تبدأ فى كيفية الاتصال و التعامل مع قواعد البيانات مثل أكسس , MySQl , java db وفى الجزء الاخير من الكتاب يعطيك فكرة عن كيفية برمجة تطبيقات الجوال من خلال NetBeans

http://egy-tech.forumegypt.net

2/14/2014



المحتويات

الباب الثانى :واجهة المستخدم الرسومية

الباب الثالث قواعد البيانات

الباب الرابع : برمجة تطبيقات الجوال

الباب الاول : الاساسيات

الفصل الاول : إعداد بيئة العمل

أخوانى الكرام أهلا وسهلا بكم فى اول درس من دروس دورة الجافا الرسومية و التى يقدمها منتدى مصر التقنى لكل متهم بالجافا ...والدروس ان شاء الله تعالى ستكون سهلة و سلسة ليس لكل مبتدىء فى الجافا فقط بل لكل من ليس لدية أى خبرة فى البرمجة و لم يتعامل مع أى لغة برمجة أخرى.... ولكن عليك قبل الشروع في تعلم لغة الجافا أن تكون متأكدا من رغبتك في تعلمها ... فمن ليست لديه الرغبة فلن ينجح ولن يكمل .. ومن عنده الرغبة ولو كان أقل نشاطا من سابقه فسوف يجد سهولة في تعلم اللغة...

وقبل المضى قدما فى شرح هذا الدرس أحب أن أنوه أنه ليس هناك ما يسمى بلغة الجافا الرسومية و الجافا غير الرسومية و انما ما قصدته من العنوان) دورة الجافا الرسومية (أننا سنقوم بإنشاء برامج ذات واجهة مستخدم رسومية عن طريق الجافا...كما يجب عدم الخلط بين الجافا والجافا سكريبت فرغم تشابة اللغتين في السياق والتراكيب إلا أنهما لغتان مختلفتان ولهما أهداف مختلفة ....أما من ناحية أخرى أريد أن أهمس في أذن كل واحد منكم....أننى لن أحدثكم عن مميزات تلك اللغة...ولن أقول لكم من هو مؤسسها ..ولن أشرح لكم مراحل تطورها ..ولن أقارن بينها وبين أي لغة برمجة أخرى ..فأنتم تعلمون قبلي جميع هذه التفاصيل .. وأنا متأكد أن أغلبكم قد سئم منها لكثرة ما تعاد وتكرر في منتديات أخرى.....

التعرف على إصدارات الجافا

و قبل البدء فى هذا الدرس هو إعطائك نبذة مختصرة عن أنواع أو إصدارات الجافا و فيما يستخدم كل إصدار و ذلك على النحو التالى:

JAVA SE : الإصدار القياسي من الجافا وهو أختصار لكلمة (Java Standard Edition) هو الجزء الخاص ببرمجة برامج سطح المكتب و الذى نحن بصدده(STANDALONE APPLICATIONS)

JAVA EE : اصدار الإنتربرايس بالإنجليزية(Java Enterprise Edition) هو يختلف عن الإصدار القياسي في أنه خاص ببرمجة الخوادم والتطبيقات الموزعة وبرمجة الويب

JAVA ME : اصدار المايكرو بالإنجليزية (Java Micro Edition) وهو اصدار خاص بالرمجة على الهواتف المحمولة والأجهزة المحدودة في المصادر عموما وتعتبر الخيار الأول لمبرمجي ألعاب الهواتف المحمولة.

كيفية إعداد جهازك لانشاء تطبيقات الجافا

لكى تتمكن أخى الكريم من إنشاء تطبيقات بلغة الجافا يجب عليك أولا ثبيت بيئة العمل الخاصة بلغة الجافا على جهازك و هى تسمى JAVA DEVELOPMENT KIT وتختصر الى **JDK** وتم تطوير تلك البيئة من قبل شركة SUN MICROSYSTEMS المملوكة لشركة أوراكل حاليا

وبعد تثبيت JDK على جهازك تستطيع لكتابة أكواد الجافا بمحرر نصوص بسيط مثل Notepad المرفق مع الويندوز .أما أنظمة التشغيل الاخرى فهي أيضاً تحتوي على محرراتها فعلى سبيل المثال هناك VI على UNIX، وكذلك علىlinux ، هناك أيضاً Text edit على الماكنتوش...

فبواسطة هذا البرنامج الصغير (المحرر) يمكنك كتابة برنامج جافا بأكمله بجميع أكواده، بعد ذلك يجب عليك حفظ هذا الكود في ملف بإمتداد.File.java :

لكن هناك مشكلة اساسية لن يحلها هذا المحرر البسيط؛ وهي أنك إن أردت كتابة برنامج كبير، فلن يتحقق لك ذلك إلا بمشقة الأنفس ! لماذا ؟ .. ببساطة لأن المحرر الذي يأتي مثلاً مدمجاً في الويندوز لا يحتوي على ألوان تفرق لك كلمات الكود لتسهيل قراءته وكذلك لتسهيل كتابته، ولا يحتوي على نوافذ متعددة تمكنك من كتابة أكواد عديدة في نفس النافذة .

كما أنه لن يظهر لك العناصر الرسومية مثل مربعات النصوص و زر الامر ...الخ لكى تقوم بوضع هذه العناصر على النافذة بكل سهولة عن طريق السحب و الافلات مما يجعل كتابة برنامج كبير أمر شبه مستحيل!

إذاً ماهو الحل؟؟

الحل ببساطة هو اننا نحتاج بيئة التطور المتكاملة Integrated Development Environement ويتم أختصارها الى IDE والتى تسهل علينا كتابة الاكواد لانشاء زر أمر مثلا أو حتى أنشاء نافذة فعن طريق خطوة واحدة يتم فيها انشاء ما تريدة من العناصر الرسومية عن طريق السحب و الافلات كما تعودنا فى الفجوال بيسك ....

ومن أمثلة تلك البيئات.JBulide / Eclipse/ netbeans ونحن هنا في هذه الدورة سوف نتكلم عن واحد من أقوى بيئات التطوير (IDE) وهو برنامج نت بينز...(netbeans)وذلك لكونه مدعوم من قبل الشركة المنتجة للغة الجافاORACLE

لذلك أرجو أن تتبع الخطوات التالية لتحميل البيئة الخاصة بلغة الجافا JDK بالاضافة الى بيئة التطور المتكاملة وهو برنامج: NETBEANS

1_الدخول على الموقع التالى

http://www.oracle.com/technetwork/java/javase/downloads/index.html

2_سنقوم بأختيار الخيار الثالث وهو تحميل JDK + NETBEANS كما فى الصورة التالية:

Java SE Downloads

Latest Release | Next Release (Early Access) | Embedded Use | Previous Releases

Java — DOWNLOAD — Java Platform (JDK) 7u4

JavaFX — DOWNLOAD — JavaFX 2.1

NetBeans — DOWNLOAD — JDK 7u4 + NetBeans

Java Java EE — DOWNLOAD — JDK 7u3 + Java EE

2_ ستظر لك الصفحة التالية... قم بالموافقة على شروط الترخيص ثم بعد ذلك حدد نظام التشغيل على جهازك و سيتم بدء عملية التحميل....

JDK 7u4 with NetBeans 7.1.2

This distribution of the JDK includes the NetBeans IDE, which is a powerful integrated development environment for developing applications on the Java platform. Learn more

You must accept the JDK 7u4 and NetBeans 7.1.2 Cobundle License Agreement to download this software.

○ Accept License Agreement ◉ Decline License Agreement

لابد من تنشيط هذا الخيار للموافقة على شروط الاستخدام

Java SE and NetBeans Cobundle (JDK 7u4 and NB 7.1.2)		
Product / File Description	File Size	Download
Linux (32-bit)	165.88 MB	jdk-7u4-nb-7_1_2-linux-ml.sh
Mac OS X v10.7 (64-bit)	160.43 MB	jdk-7u4-nb-7_1_2-macosx-ml.dmg
Solaris SPARC (32-bit)	167.65 MB	jdk-7u4-nb-7_1_2-solaris-sparc-ml.sh
Solaris x86 (32-bit)	158.86 MB	jdk-7u4-nb-7_1_2-solaris-x86-ml.sh
Windows (32-bit)	176 MB	jdk-7u4-nb-7_1_2-windows-ml.exe

3_ أما اذا لم يتم بدء عملية التحميل وظهرت الرسالة التالية...

Thank you for accessing Oracle Technology Network.

In compliance with U.S. and applicable Export laws we are unable to process your request. Please contact RPLS-Ops_ww@oracle.com if you believe you are receiving this notice in error.

- Back to Previous Page
- Site Map
- Product Index
- Software Download Index
- To **search** for your page, try our Search function.

search site | Technology Network | Refine Search

فمعنى ذلك أن الدولة التى تقطن فيها عليها عقوبات مفروضة من قبل الولايات المتحدة الامريكية مثل السودان و سوريا فأعلم أنك لن تستطيع التحميل من هذا الموقع ...لذلك سيقوم منتدى مصر التقنى برفع JDK + NetBeans على سرفر خاص حتى يتثنى للجميع تحملية.

4-بعد التحميل سيظهر لك ملفان ...الاول خاص ببيئة تشغيل الجافا على جهازك... JDK أما الثانى هو الخاص ببرنامج... NetBeans كما فى الصورة التالية:



تثبيت JDK البيئة الخاصة بلغة الجافا

يجب عليك أخى الكريم تثبيت JDK أولا قبل تثبيت برنامج NetBeans إن لم يحدث غير ذلك سوف يظهر لك رسالة خطأ عند تثبيت NetBeans مفداها أنه لايوجد على جهازك بيئة الجافا.JDK

وكيفيك لتثبيت JDK الضغط على زر Next حتى يتم الانتهاء من التثبيت ....لكن بقى شىء أخير يجب فعله ....أتبع التالى:

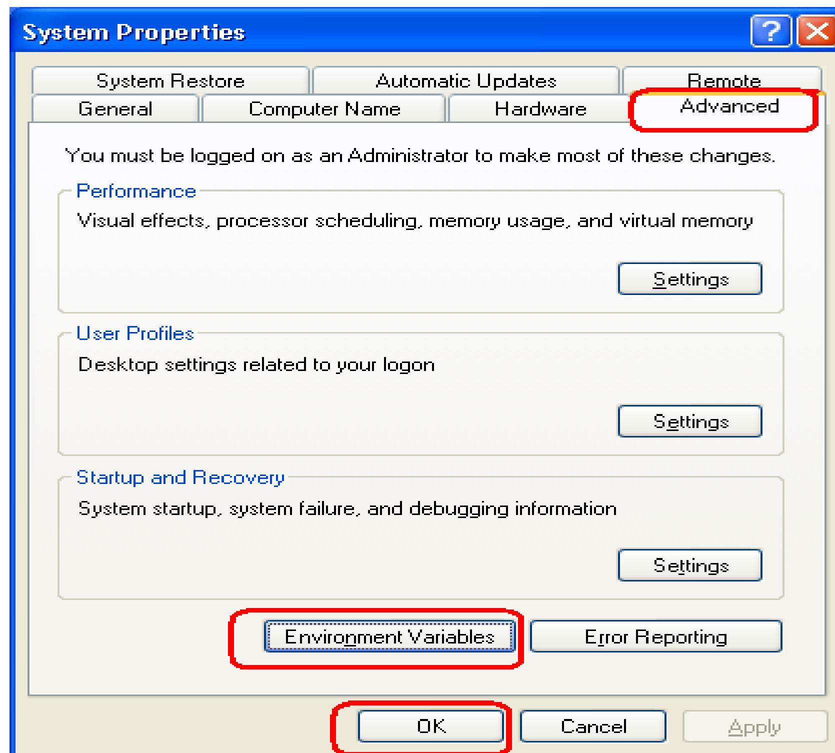
1_ الدخول على المجلد الذى تم تثبيت فيه..... JDK فمثلا أنا قمت بالتثيت على القرص C و بالتالى فأن مجلد التثبيت سيكون فى مجلد C:\Program File

2_ عند الدخول فى هذا المجلد.... C:\Program Files ستجد مجلد بأسم... JAVA قم فتح هذا المجلد ....ستجد فيه مجلدان أخران ....واحد يسمى JDK1.7.0_01 و الاخر يسمى JRE7

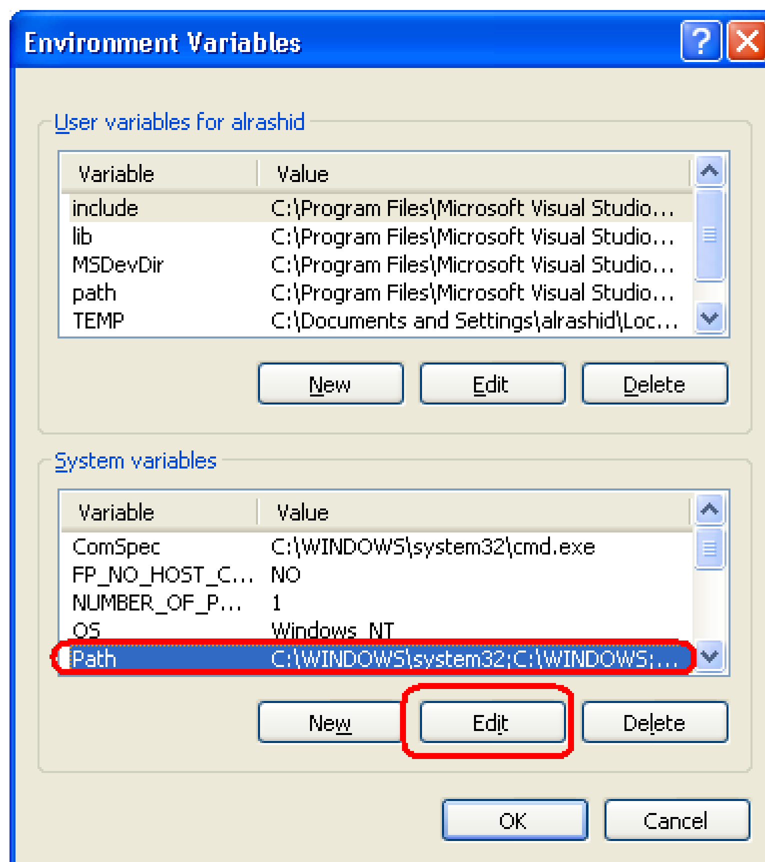
3_قم بفتح المجلد الذى يسمى.. jdk1.7.0_01 ستجد به مجلد يسمى.BIN قم فتحه هو ايضا ...وعندما تكون بداخلة أنسخ المسار الموجود فى شريط العنوان كما فى الصورة التالية:



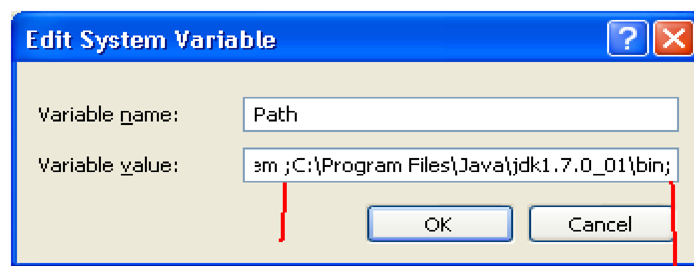
4_ثم بعد ذلك أذهب الى My Computer ثم كليك شمال ستظهر لك نافذة الخصائص ....أضغط على التويب... Advanced ستجد فى أخر النافذة زر الامر.... Envirment Variables كما فى الصورة التالية:



5_ ستظهر لك النافذة التالية ....كل ما عليك هو تحديد path ثم بعد ذلك أضغط Edit كما فى الشكل التالى :



6_ ستظهر لك نافذة صغيرة ...فى مربع النص VARIABLES VALUE ستجد عدة سطر كلمات ....كل ما عليك أن تضع مؤشر الكتابة نهاية هذا السطر ...وعندما تتأكد أن المؤشر فعلا نهاية السطر قم بوضع فاصلة منقوطة ; ثم ضع المسار الذى قمنا بحفظة سابقا.(C:\PROGRAM FILES\JAVA\JDK1.7.0_01\BIN) ثم ننهى السطر أيضا بالفاصلة المنقوطة... ; كما فى الصورة التالية :



لاحظ أن الجملة بدأت و أنتهة بالفاصلة المنقوطة

7_ثم بعد ذلك أضغط OK _OK_OK _OK الى أن تنهى جميع النوافذ عندك ....وهكذا تكون قد جهزت بيئة عمل الجافا بنجاح على جهازك.......

و أخير قم بتثيت برنامج... NETBEANS ويكفيك لذلك الضغط على زر NEXT حتى الانتهاء من التثبيت ....

الفصل الثانى : الهيكل العام لبناء تطبيقات الجافا

أهلا وسهلا أخوانى أعضاء و زوار منتدى مصر التقنى فى ثانى درس من دروس دورة الجافا الرسومية بأستخدام NetBeans ....اليوم إن شاء الله تعالى سنتعرف سويا الهيكل العام لبناء أى تطبيق جافا...وقبل البدء فى الشرح يجب أن نقف على بعض المصطلحات فى الجافا .

كل برنامج فى لغة الجافا مكون من فئة CLASS أو أكثر ....أذن ما هى الفئة CLASS ؟

الفئة او الكلاس (Class) في البرمجة كائنية التوجه (Object-Oriented Programming)، تمثل المخطط الأولي لإنشاء الكائنات (Objects)، حيث يحتوي هذا المخطط على مجموعة الخصائص (Attributes) والوظائف (Methods)، التي سوف تتشارك فيها جميع الكائنات التي سيتم إنشاءها من هذه الفئة :

وهناك نوعين من الفئات الكلاس :

1_فئة أساسية أو رئيسة: main يجب توفرها في أي برنامج.

2 _فئة ثانوى يتم استعمالها من طرف الكلاس الاساسى main، وهي تحتوي بشكل عام على الوظائف التي يجب على البرنامج القيام بها .

ولكى يتم فهم ما تم ذكرة تعالوا نتعرف على الشكل العام لبناء أى برنامج فى الجافا .

الهيكل العام لبناء أى تطبيق جافا

بعدما مررنا بتعريف بسيط للفئة Class و معرفة أنواعها وقلنا أن أى تطبيق فى جافا لابد أن يحتوى على فئة Class على الاقل ، ندخل مباشرة لمعرفة الهيكل العام لبناء تطبيق الجافا...



شرح الهيكل العام لبناء أى برنامج جافا .

- **فى السطر الاول** يتم إنشاء كلاس Class ...وذلك عن طريق الامر Public ثم يتبعة الكلمة Class وهى من الكلمات المحجوزة فى لغة الجافا والتى تقوم بتعريف أو أنشاء الفئة ....ثم يتبعها أسم الفئة ولقد تم تسميته هنا بالاسم الذي نريده Welcome
- **السطر الثانى** : تم فتح { قوس بداية الفئة Welcome أو كلاس Welcome .
- **السطر الثالث** : public static void main(String args[ ]) ....فهو يمثل جزءا من كل تطبيق جافا (Java Application) حيث يبدأ تنفيذ البرنامج من هنا أى أن هذا السطر هو نقطة بداية البرنامج ...معنى ذلك أن أى تطبيق بلغة الجافا لابد و أن يحتوى على نقطة البداية ولا لن يتم تنفيذ شيئا .....ويتم أنشاءة كما فى الشكل السابق
- **السطر الرابع** : تم فتح { قوس بداية الدالة main
- **أما السطر الخامس** فهو فارغ لوضع كود برمجى أى ما تريد أن يقوم به البرنامج مثل طباعة سطر معين ، حساب عدد معين ....الخ

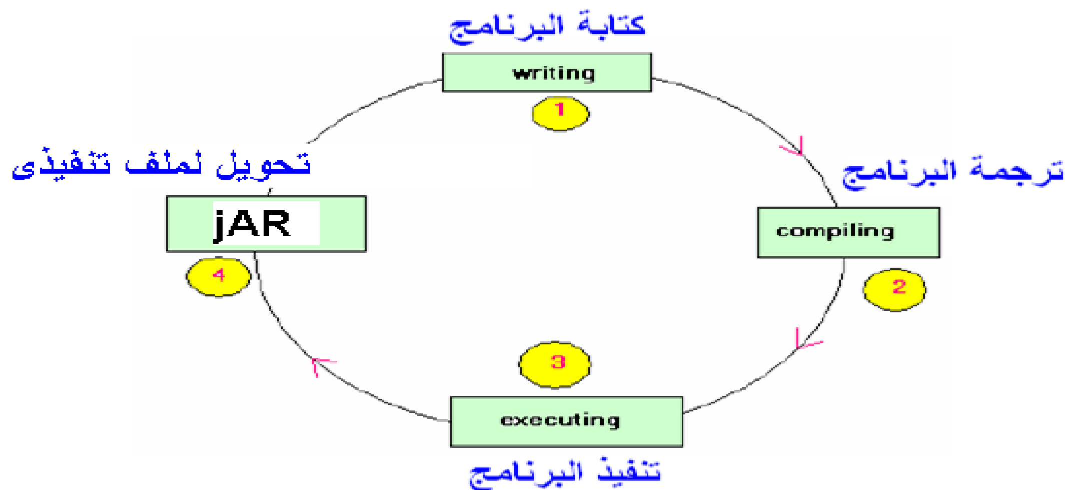
بعد ذلك يتم إنهاء البرنامج بقوسي النهاية حيث يمثل القوس الأول نهاية الوظيفة method للدالة main والقوس الآخر يمثل نهاية الفصيلة class

ملاحظات هامة :

- لابد وأن تنتهي كل جملة (سطر أو أمر) بعلامة الفاصلة المنقوطة ;
- عند تعريف الفئة class يجب أن يبدأ بكلمة public .
- يجب أن يبدأ أسم الفئة class بحرف لا برقم ،ولا يحتوى على أى مسافات فارغة ولا يكون أحد الاسماء المحجوزة فى اللغة
- يفضل أن يبدأ أسم CLASS بحرف كبير مثل Welcome، كذلك إذا كان يتكون من أكثر من أسم فإن كل أسم يبدأ بحرف كبير مثل SampleClassName .

خطوات تنفيذ أى تطبيق جافا

بعد ما تعرفنا على الهيكل العام لانشاء تطبيق جافا ...نأتى الان الى مراحل أو خطوات تنفيذ التطبيق :



- **الخطوة الاولى** : كتابة أكواد البرنامج ....اين ؟....بعيدا عن بيئة NetBeans سنتحدث أولا عن كيفية كتابة أكواد الجافا فى محرر نصوص عادى مثل Notepad و إنشاء تطبيق جافا بواسطته ...ثم بعد ذلك ننشىء نفس التطبيق بواسطة NETBEANS لنرى كيف ساعدنا فى كتابة الكود و انشاء التطبيق بكل سهولة و يسر ...
- **الخطوة الثانية** :ترجمة البرنامج ....لماذا ؟ ...بمعنى تحويل وترجمة شيفرات مكتوبة بلغة معينة أو source code إلى لغة يفهمها الحاسوب و نظام التشغيل...وذلك لكون كل ما كتبته من أكواد هو عبارة عن نص لا يفهمه الحاسوب، وانما تفهمه أنت ...وفي حالة لغة جافا يقوم المترجم بترجمة النص المكتوب إلى ما يسمى بالبايتكود bytecode حتى يفهمة الحاسب.
- **الخطوة الثالثة** :تشغيل و تنفيذ البرنامج لنرى مخرجات برنامجنا
- الخطوة الرابعة : تحويل برنامجنا الى ملف تنفيذى لنستطيع نقل وتشغيل برنامجنا على الاجهزة الاخرى فى حالة الجافا سيكون برنامجنا فى صورته النهاية بأمتداد JAR

تطبيق عملى بأستخدام محرر نصوص بسيط NOTEPAD

اكتب برنامج يقوم بأظهار رسالة ترحيب ...؟؟؟؟؟

الحل

الخطوة الاولى: كتابة التطبيق .... نقوم بفتح محرر النصوص NOTEPAD ...وكتابة الكود التالى فيه :

```
import javax.swing.JOptionPane;

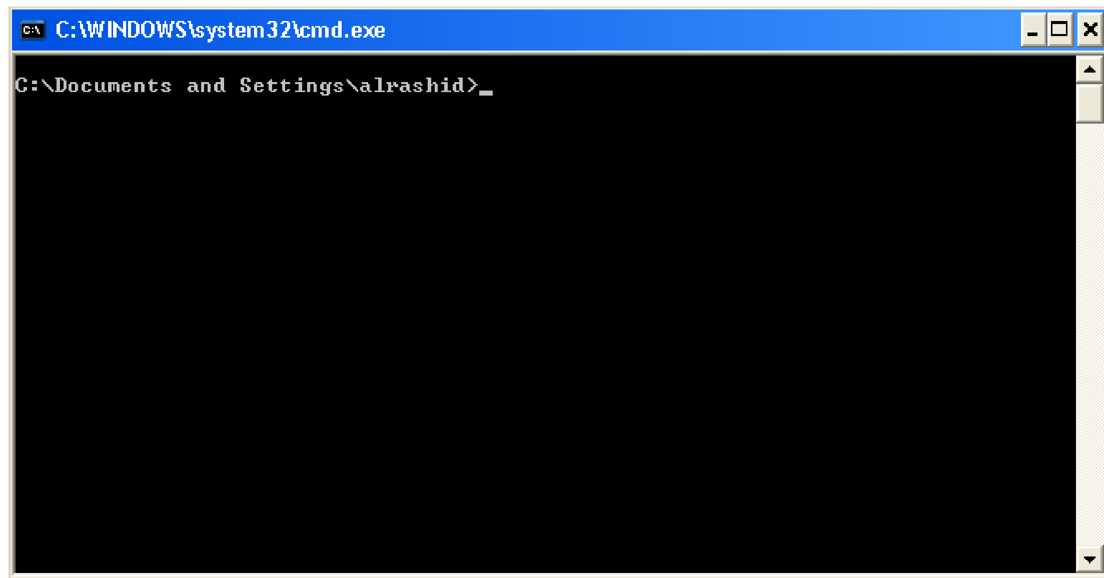
 public class Welcome
 {
 public static void main( String args[ ] )
        {
JOptionPane.showMessageDialog(null, "" );
        }
 }
```

شرح الكود السابق :

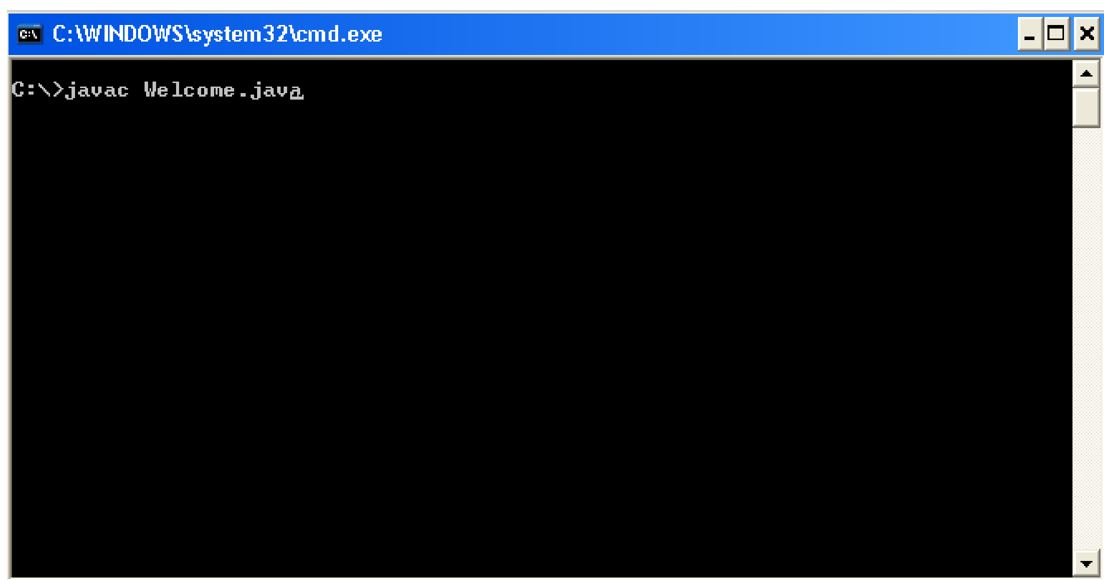
- ❖ السطر الاول ...قمنا بأستدعاء وإستيراد المكتبة swing _وهى الخاصة بالعناصر الرسومية وسوف يتم شرحها بالتفصيل فى دورس لاحقة_ وحددنا العنصر الرسومى JOptionPane وهو المستخدم بعرض صناديق الحوار_ ولها درس أخر منفصل _ وكل ما نريدة الان هو كيفية أستيراد العنصر الرسومى ...
- ❖ السطر الثانى ...أنشأنا CLASS يسمى Welcome
- ❖ السطر الثالث ...قوس { للتعبير عن بداية الكلاس Welcome
- ❖ السطر الرابع ...أنشأنا الدالة الرئسية main للبرنامج أو نقطة البداية و التى سيبدأ من خلالها الحاسب بتنفيذ الاوامر
- ❖ السطر الخامس ...قوس { للتعبير عن بداية الدالة MAIN
- ❖ السطر السادس ...كتبنا المطلوب تنفيذة من برنامجنا هو أظهار رسالة للمستخدم عن طريق الامر JOptionPane.showMessageDialog ..تحتوى على النص_ مرحبا بكم فى منتدى مصر التتقى _ويجب ملاحظة أننا فتحنا قوس ( ) ثم كتبنا الكلمة المحجوزة فى لغة الجافا NULL بعد ذلك وضعنا فاصلة ، ثم علامتى أقتباس بينهما النص المراد عرضة ..و أنهينا السطر بالفاصلة المنقوطة ؛

الان نقوم بحفظ الملف لكن بأمتداد JAVA . وذلك عن طريق SAVE AS ولابد أن يكون أسم الملف بنفس أسم CLASS وسوف يعطى المترجم خطأ عند الترجمة اذا لم يكن أسم الملف هو نفس أسم CLASS وكذلك إن لم يكن بأمتداد JAVA .

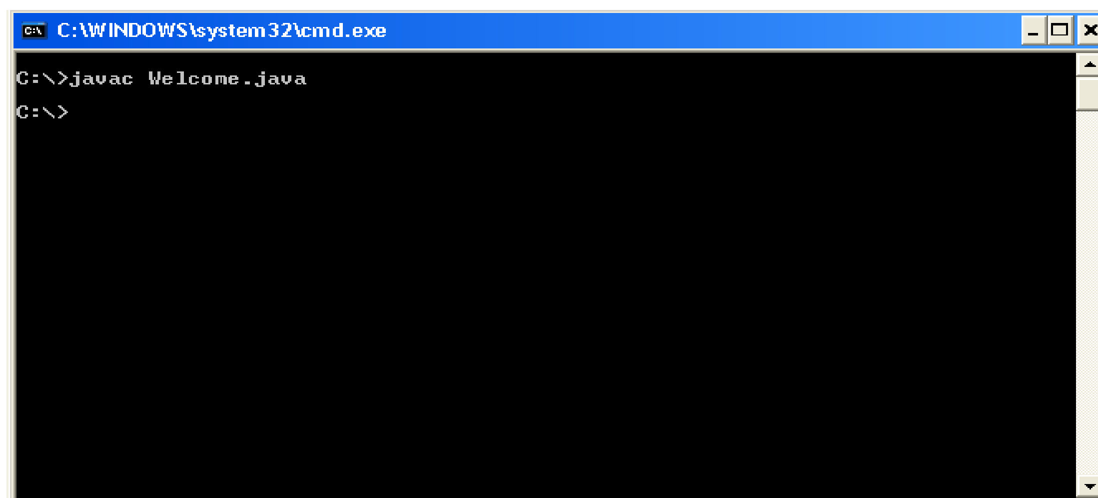
الخطوة الثانية : ترجمة البرنامج ...بعد ما قمنا بكتابة التطبيق و حفظناه بأمتداد JAVA. بنفس أسم الكلاس الرئيسى WELCOME ..لنأتى الان لمرحلة الترجمة ...ونستطيع عمل ذلك من خلال نافذة الاوامر command lin ...نذهب START ثم RUN ثم نكتب الامر CMD ثم أوكية ....فتظهر لنا نافذة الاوامر ...كما بالشكل التالى .



أحدد المسار الذى حفظ فية التطبيق Welcome.java ...فى مثالنا هذا قمت بحفظة على C ...بعد ذلك سنقوم بكتابة الامر javac Welcome.java كما فى الشكل التالى :



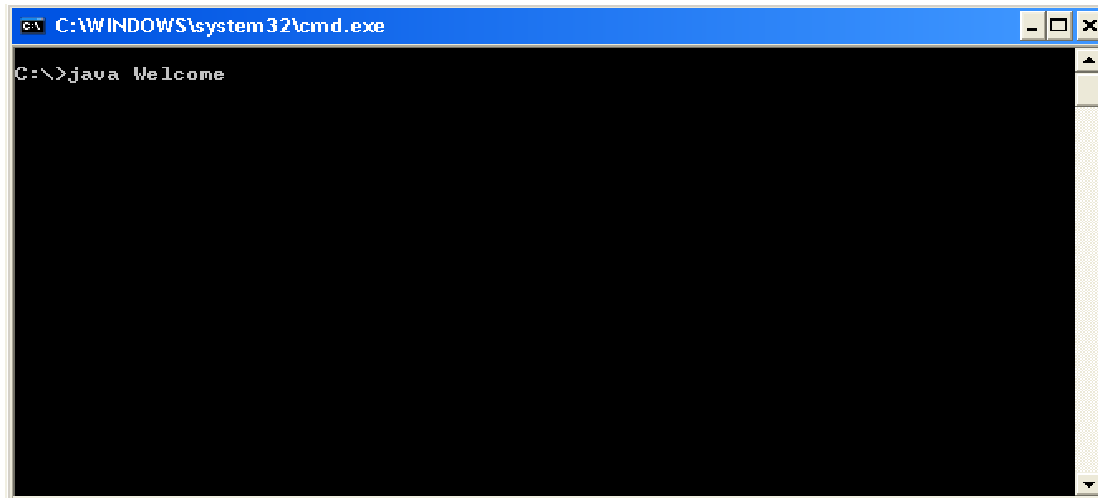
ويجب ملاحظة أننا كتبنا فى نافذة الاوامر **جافا سى** ثم مسافة ثم أسم التطبيق بالامتداد Welcome.java ...الان قم بالضغط على زر Enter لتنفيذ الامر ...أن ظهرت الشاشة التالية بدون أى ERRO أو سطور فمعنى ذلك أن البرنامج قد تم ترجمته بصورة صحيحة .كما فى الصورة التالية :



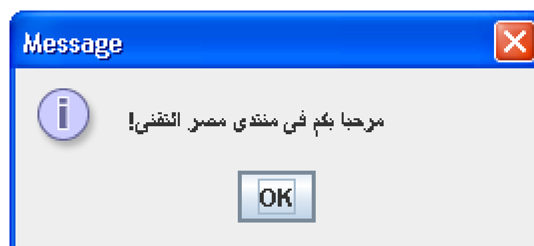
ولكى تتأكد أن التطبيق قد تم ترجمتة ...أذهب الى الملجد C ستجد ملف بجوار الملف Welcome.java يسمى Welcome.class كما فى الصورة التالية :



الخطوة الثالثة : تنفيذ و تشغيل البرنامج ....كل ما عليك فعلة الان لكى تتمكن من تشغيل التطبيق هو كتابة الامر JAVA WELCOME ....كما فى الصورة التالية :



الان قم بالضغط على زر ENTER ....ستجد برنامجك يعمل جيدا و تظهرك رسالة كما فى الصورة التالية :



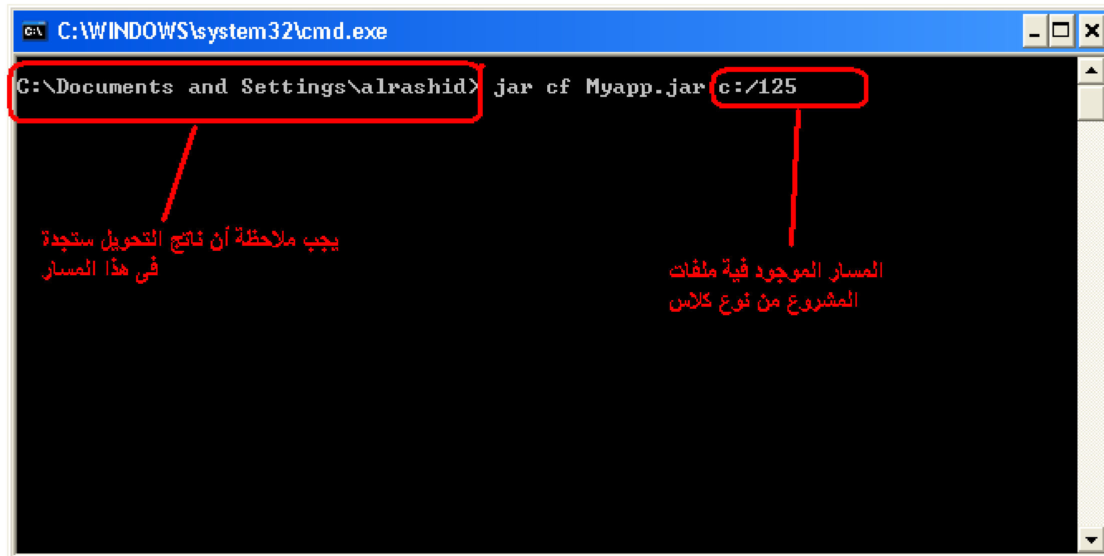
الخطوة الرابعة و الاخيرة ...تحويل التطبيق الى ملف JAR

بعد الانتهاء من كتابة أكواد التطبيق و ترجمتة و تشغلية ....يقى لك خطوة أخيرة لكى يعمل البرنامج بشكل صحيح على الاجهزة الاخرى هو تحويله الى ملف تنفيذى من نوع JAR .....ويتم ذلك عن طريق نافذة الاوامر وذلك بكتابة الامر التالى :

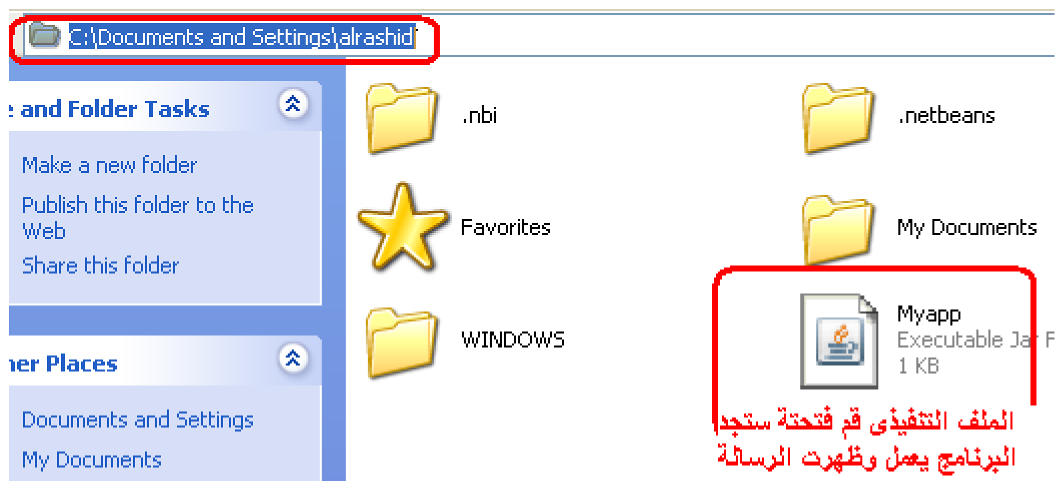
JAR CF JAR_FILE_NAME DIRECTORY_NAME

ويجب ملاحظة أن ما هو مكتوب باللون الاحمر و الازرق مجرد متغيرات لشرح الامر فقط ونستطيع تغيرها JAR_FILE_NAME... هو لوضع أسم للتطبيق بأمتداد JAR ونستطيع وضع أى أسم بغض النظر عن أسم CLASS ....أما ال DIRECTORY_NAME هو وضع مسار مجلد التطبيق ...فمثلا فى برنامجنا هذا سأقوم بكتابة الامر التالى :

JAR CF MYAPP.JAR C:/125



هنا قمت بكتابة الامر JAR CF ثم وضعت أسم لبرنامجى هو Myapp.jar بالامتداد ثم وضعت المسار الذى يتواجد فية ملفات المشروع و هو فى مجلد ١٢٥ على C .... النتجة أننا سأجد ملف myapp فى المسار الذى كنت متواجد فية فى نافذة الاوامر عندما قمت بأنشائة كما فى الصورة التالية :



و الى هنا ننتهى من الجزء الاول من هذا الدرس أما فى الجزء الثانى سنقوم بإنشاء نفس المشروع لكن فى بيئة NETBEANS لنرى كم الجهد الذى سيوفره لنا

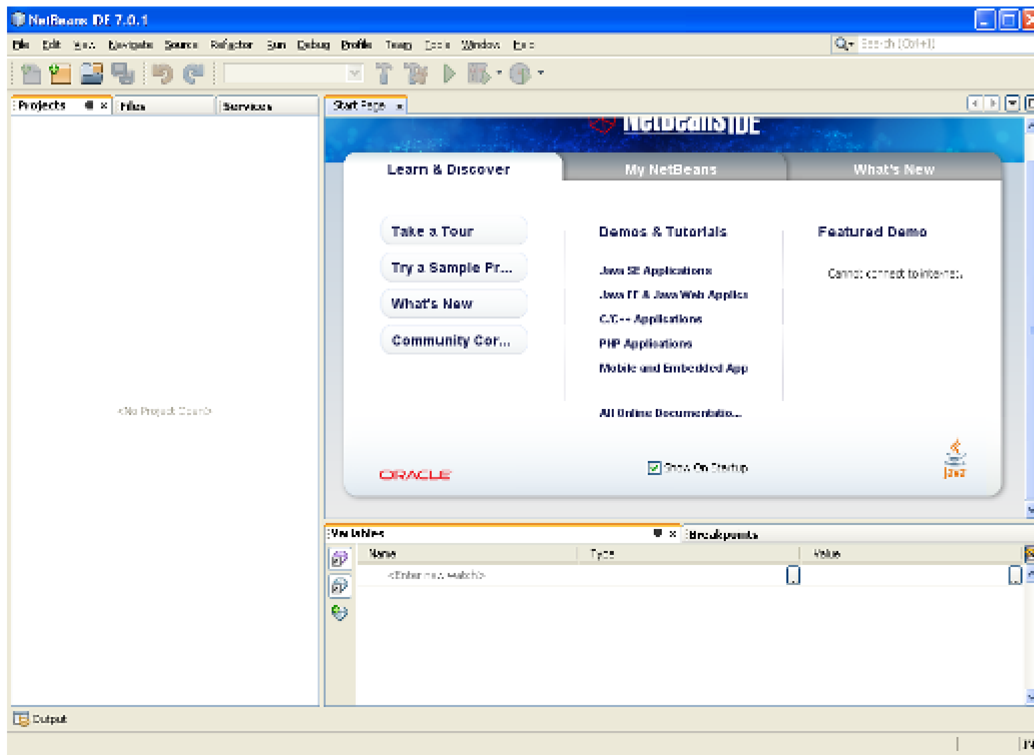
إنشاء أول تطبيق من خلال NetBean

تعرفنا فى الجزء الاول على كيفية بناء تطبيق جافا ...وقلنا أن أى تطبيق جافا يتكون من كلاس أو أكثر ...و طبعا عرفنا ماذا يعنى ماذا يقصد بالكلاس ويجب أن يحتوى على نقطة البداية و هى الدالة main .

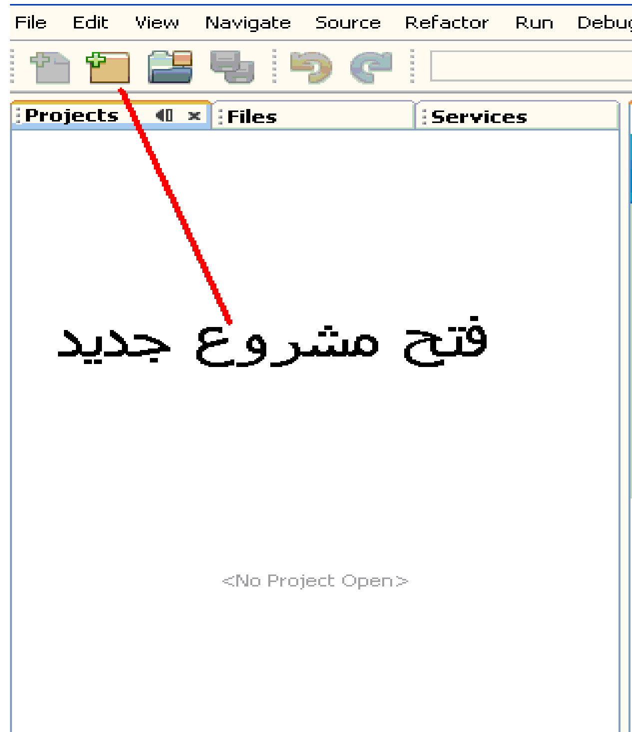
و تعرفنا سويا على خطوات إنشاء تطبيق جافا من كتابة الكود فى محرر بسيط Notepad ثم حفظ الملف بأمتداد JAVA.و تعلمنا أنه لابد من ترجمة الملف بأستخدام الامر JAVAC عن طريق نافذة الاوامر ...ثم بعد ذلك قمنا بتشغيله عن طريق الامر JAVA ...واخيرا قمنا بحتويلة الى ملف تنفيذى JAR لكى يعمل بمجرد الضغط علية و نقله على الاجهزة الاخرى ..

أما فى هذا الجزء سنتعلم سويا كيفية إنشاء نفس التطبيق ولكن بأستخدم NetBeans لذلك ارجوا أن تتبع الخطوات التالية :

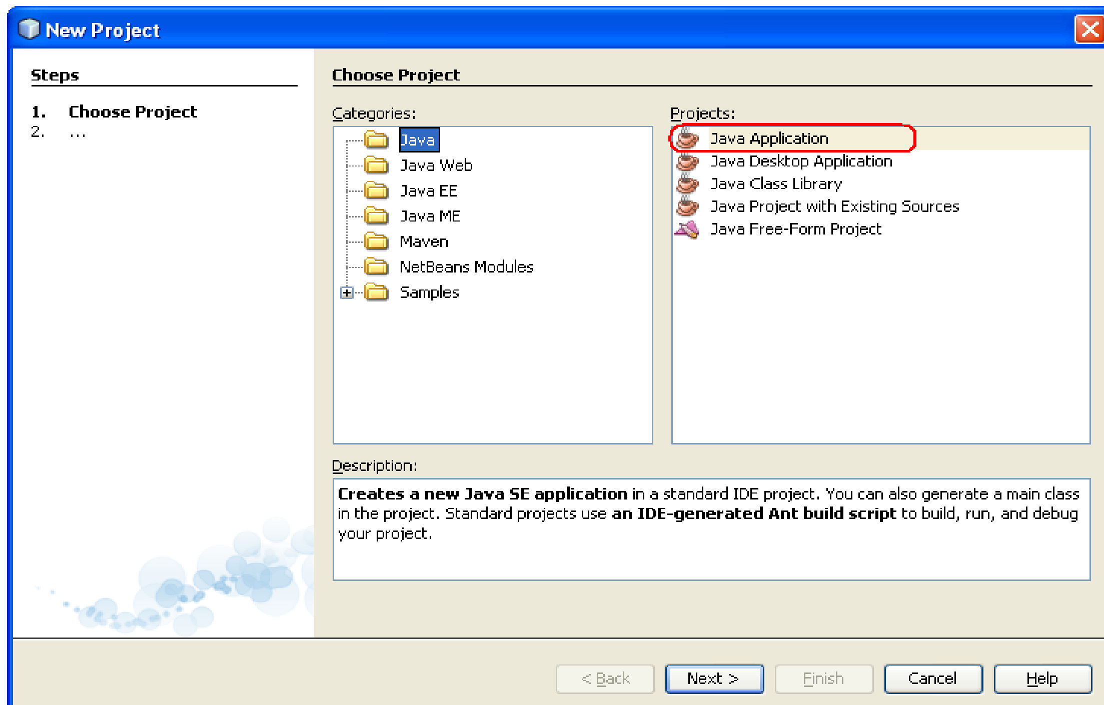
الخطوة الاولى : قم بتشغيل NetBeans ....ستجد واجهة البرنامج كما فى الشكل التالى :



الخطوة الثانية: الضغط على أيقونة إنشاء مشروع جديد .....كما فى الصورة التالية :

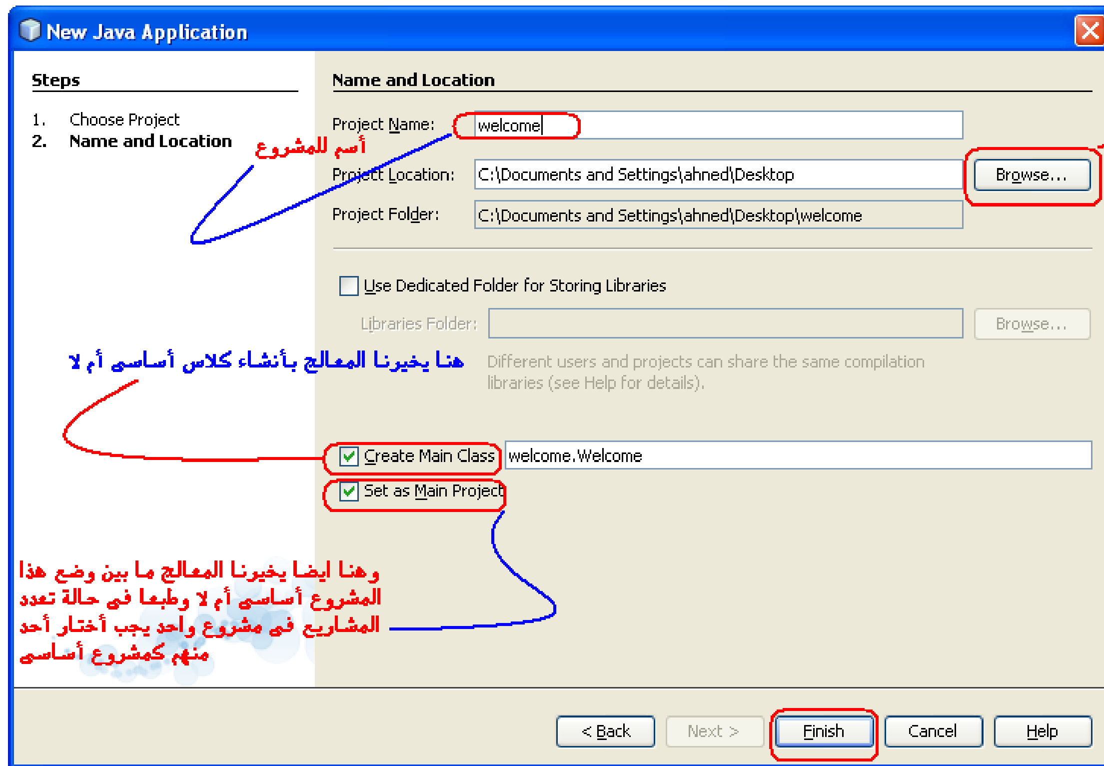


سوف يظهر لنا مربع حوارى لاختيار نوع المشروع ....كما فى الصورة التالية :



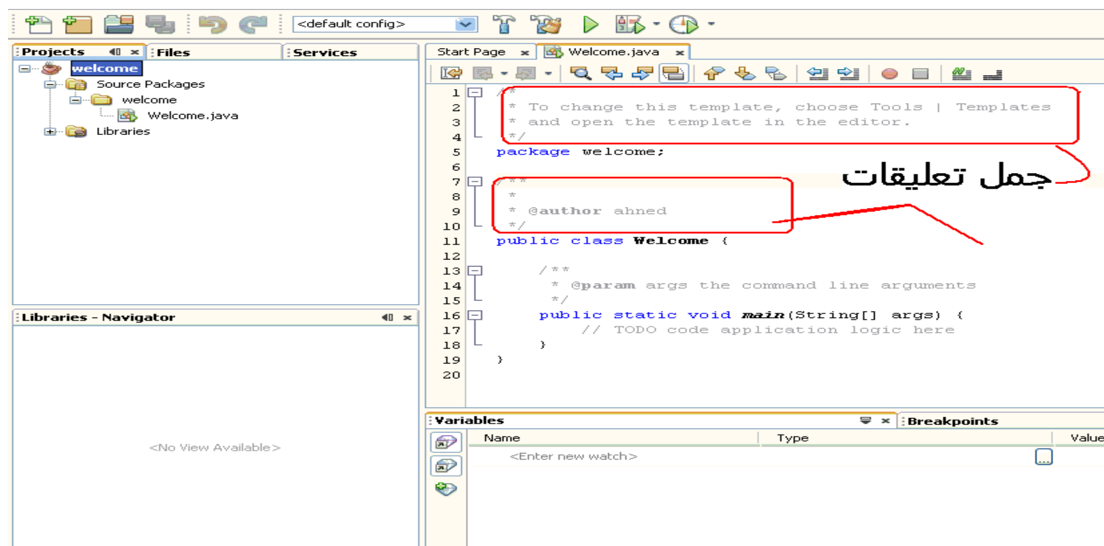
ستلاحظ أخى الكريم من الصورة السابقة فى التبويب Categories أنواع من الجافا فمنها java EE وقلنا أنه خاص ببرمجة الخوادم والتطبيقات الموزعة و Java ME و هو الخاص ببرمجة تطبيقات الموبايل ...وسوف نتحدث عنه فى

دورس متقدمة .....الذى يهمنا الان هو التصنيف JAVA و عند الضغط عليه ستجد به عدة مشاريع .. JAVA application و java Desktop application .....الخ ....قم بأختيار مشروع من نوع java application كما فى الصورة السابقة ثم أضغط NEXT .... ستجد المربع الحوارى التالى :



الخطوة الثالثة: وضع أسم للمشروع كما فى الصورة السابقة ....وستجد ايضا أن المعالج يحدد لك المسار الذى سيتم وضع ملفات المشروع فية ولكن تستطيع تغير هذا المسار كما تريد من خلال زر Browse ...أكتب أسم للمشروع فى الخانة المخصص له ويجب أن يكون بالانجليزية .....

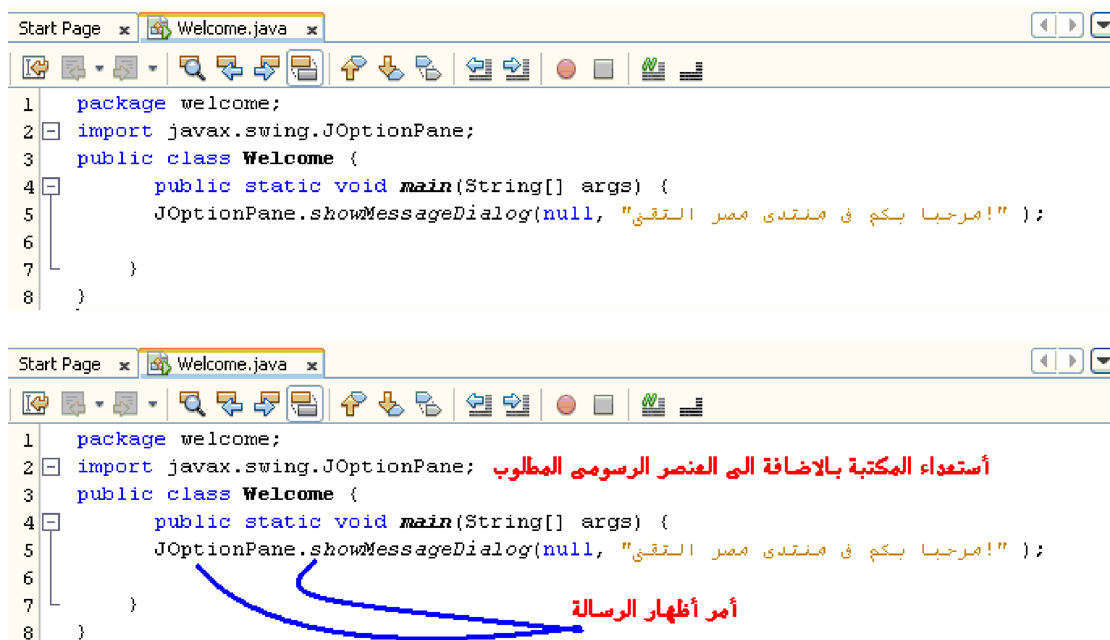
.فى نهاية المعالج ستجد مربعين أختيار الاول: create main class ....وهو الخاص بأنشاء الكلاس الرئيسى لبرنامجنا سنقوم بتحديدة .....أما الخيار الثانى: set as main project وطبعا هذا فى حالة وجود عدة مشاريع ....فى حال مثالنا هذا سواء قمنا بتحديدة أم لم نقم بذلك فلا فائدة فيه وذلك لكون التطبيق الذى نحن بصددة مكون من مشروع واحد فقط ....المهم الان قم بالضغط على إنهاء لتظهر لنا شاشة كتابة الكود كما فى الصورة التالية :



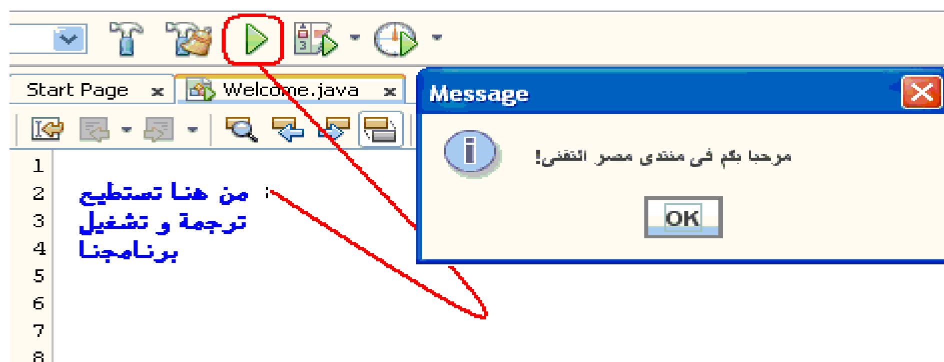
كما تلاحظون من الصورة السابقة بأن البرنامج NetBeans قد أنشىء لنا تلقائيا الكلاس welcome كما أنه أنشىء الدالة MAIN أو نقطة بداية التطبيق وكل ما علينا الان فعلة هو كتابة المطلوب تنفيذة من التطبيق و هو إظهار رسالة الترحيب .

أما ما هو مكتوب بالخط الرمادى مجرد جمل تعليق مثل الجمل الخضراء التى تجدها فى محرر الفجوال بيسك ٦ لا يلتفت اليها البرنامج أثناء التنفيذ و تستطيع حذفها إن شئت أو تستطيع كتابة جمل أخرى و ذلك بوضع علامة // فى حالة كان التعليق سطر واحد ...ووضع علامة **/ ثم جمل تعليق فى أكثر من سطر ثم أنهاء جمل التعليق بوضع العامة /* كما فى الصورة السابقة .

الخطوة الرابعة : كتابة التطبيق ....الان سنقوم بكتابة الكود الخاص باظهار الرسالة و ذلك عن طريق أستدعاء المكتبة SWING و العنصر الرسومى JOptionPane أولا ....ثم بعد ذلك أعطاء الامر لاظهار الرسالة كما فى الصورة التالية :

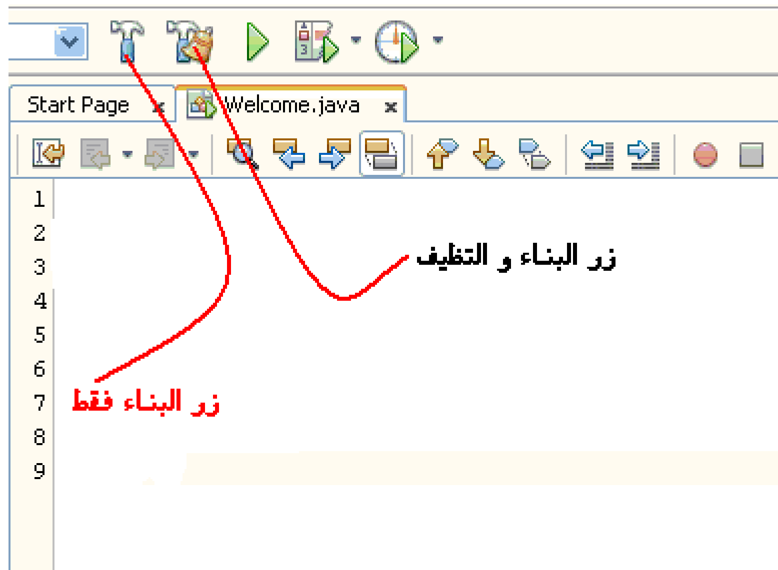


الخطوة الخامسة : تشغيل التطبيق ......بعدما قمنا بكتابة التطبيق فى بيئة NetBeans سنقوم الان بتشغيل التطبيق ليس من خلال نافذة الاوامر_ كما فعلنا فى الجزء الاول _و إنما من خلال NetBeans .....قم الان بالضغط على أيقونة RUN الموجودة فى شريط مهام NetBeans كما فى الصورة التالية :



الخطوة السادسة و الاخيرة : تحويل البرنامج لملف تنفيذى من نوع JAR ....لكى تقوم بهذة الخطوة فى بيئة NetBeans عليك أخى الكريم تشغيل التطبيق أولا ...لماذا؟ لان عندما نقوم بشغيل البرنامج يقوم NETBEANS بترجمة التطبيق ثم بعد ذلك يتم تشغيله ....و بالتالى فلو قمنا بعملية التحول تلك قبل عملية التشغيل سيكون التطبيق لم يتم ترجمتة و لن نستطيع تحويله الى ملف تنفيذى JAR .....

لنأتى الان لكيفية تحويل التطبيق الى ملف تنفيذى JAR من خلال NetBeans .....لكى نقوم بذلك نذهب الى شريط مهام NetBeans و نضغط على زر clean and build أو زر build فهذا أو ذاك يقوم بعملية بناء الشروع الى ملف تنفيذى JAR لكن الزر الاول يقوم بعملية تنظيف لبعض الملفات الغير ضروية للتطبيق ثم البناء ....كما فى الصورة التالية :



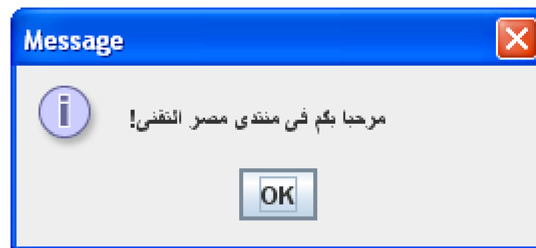
قم بالضغط على أحد منهم و أنتظر لحظات ستجد NETBEANS يقوم بعملية البناء الى ملف تنفيذى و ستجد تطبيقك كامل فى المسار الذى سيوضحة لك NETBEANS كما فى الصورة التالية :

```
Output - welcome (clean,jar)
init:
deps-clean:
Updating property file: D:\welcome\build\built-clean.properties
Deleting directory D:\welcome\build
clean:
init:
deps-jar:
Created dir: D:\welcome\build
Updating property file: D:\welcome\build\built-jar.properties
Created dir: D:\welcome\build\classes
Created dir: D:\welcome\build\empty
Created dir: D:\welcome\build\generated-sources\ap-source-output
Compiling 1 source file to D:\welcome\build\classes
compile:
Created dir: D:\welcome\dist
Copying 1 file to D:\welcome\build
Nothing to copy.
Building jar: D:\welcome\dist\welcome.jar
To run this application from the command line without Ant, try:
java -jar "D:\welcome\dist\welcome.jar"
jar:
BUILD SUCCESSFUL (total time: 4 seconds)
```

المسار الموجود فية الملف التنفيذى jar

العملية تمت بنجاح

أدخل المسار الموجود فية الملف التنفيذى للتطبيق ...ثم قم فتحتة و تشغيلة من خلال الضغط علية ستجد الرسالة قد ظهرت بهذا الشكل :



الفصل الثالث : أنواع البيانات وكيفية تمثيل الثوابت و المتغيرات

فى الدرس السابق تعلمنا سويا كيفية إنشاء تطبيق جافا عن طريق محرر النصوص NOTEPAD و ايضا عن طريق NETBEANS ...

اليوم إن شاء الله تعالى سنتكلم عن درس مهم جدا وهو عن كيفية تمثيل البيانات و العناصر الاساسية و التى تستخدم فى بناء تطبيقات الجافا مثل الثوابت و المتغيرات و غيرهما و وسوف نقوم بشرح هذه المكونات من خلال أمثلة مكتوبة بلغة الجافا ....وقبل البدء تعالوا نتذكر سويا أنواع البيانات التى ستم تمثيلها!!

أنواع البيانات فى الجافا

تحتوى لغة الجافا الحروف الأبجدية (LETTERS) سواء كانت الحروف الكبيرة (CAPITAL LETTERS) من A إلى Z وكذلك الحروف الصغيرة (SMALL LETTERS) من A إلى.Z بالضافة الى الأرقام العددية (DIGITS) من ٠ إلى ٩

وايضا الحروف الخاصة (SPECIAL CHARACTERS) وهي تلك الحروف التي ليست بأعداد أو بحروف أبجدية ولكنها تكون على هيئة رموز كالآتي(+,-,/,//,<,>,$,#,%,(),||,!,[],!=,,;,”,) :

لكن تعالى الان عزيزى القارئ تصنف تلك البيانات فى لغة الجافا حسب طولها والسعة التخزينية لها فى الذاكرة كما يلي -:

1_بيانات من النوع المنطقى: BOOLEAN

و يأخذ القيم صفر او واحد ويكون حجمة فى الذاكرة BIT ويستخدم فى حالة المتغيرات التى تكون احدى قيمتين معلومتين مثلا (نعم و لا)أو بمعنى أدق (TRUE OR FALSE) والقيمة الافتراضية فى حالة الاعلان عن متغير وعدم إعطاؤه قيمة تكون FALSE

2_بيانات من النوع: BIT

وهى البيانات التى تكون قيامها اعداد صحيحة سالبة او موجبة محصورة بين - ١٢٨ كحد أدنى و ١٢٨ كحد أقصى مثل (١٠,٥- ١٠- ١١)

3_بيانات من النوع: CHAR

وهى التى تكون قيمها عبارة عن حرف واحد او قيمة لاحدى متسلسلات الهروب او رقم وهذا الرقم يعبر عن حرف ولكن بنظام الاسكى كود حيث فى هذا النظام كل حرف يمثل برقم معين وحجمها فى الذاكرة.

4_بيانات من النوع: INT

وهى تمثل بقيم صحيحة موجبة او سالبة محصورة بين -٢١٤٧٤٨٣٦٤٨ كحد ادنى و٢١٤٧٤٨٣٦٤٧ كحد أقصى وهو أكثر أنواع البيانات أستخداما فى الجافا.

5_بيانات من النوع: FLOAT

هى تمثل بقيم الاعداد حقيقية اى اعداد بها علامة عشرية او عداد كسرية او اعداد اسية

6_بيانات من النوع : DOUBLE

وهى تمثل ايضا الاعاداد الحقيقية مثل FLOAT إلا أن الفرق هو أن البيانات من نوع DOUBLE تكون كبيرة جدا جدا

و الان أخى الكريم بعد ما تعرفنا على البيانات فى لغة الجافا و أنواعها ....تعالوا نتعلم سويا كيفية تمثلها

<u>أولا : الثوابث:</u>

هى عبارة عن قيم ثابتة يراد الاحتفاظ بها طوال فترة تشغيل البرنامج ولا تتغير قيمتها ابدا ....و تنقسم الى نوعين:

الاول ثوابت رقمية: NUMERIC CONSTANTS

هى تللك الاعداد ذات القيم الصحيحة والحقيقية التى تحوى تسلسلا من الارقام سواء كانت اعداد سالبة او اعداد موجبة او صفر ....وهى بدورها تنقسم الى :

- **ثوابت رقمية صحيحة.... INTEGER** وهى التى تحتوى على ارقام سواء كانت سالبة أو موجبة لكن لا تحتوى على كسور أو علامة عشرية
- **ثوابت رقمية حقيقية .. FLOATING CONSTANT** وهى تلك الاعداد السالبة أو الموجبة لكنها تحتوى على كسور أو علامة عشرية

الثانى ثوابت رمزية: NON-NUMERIC CONSTANTS

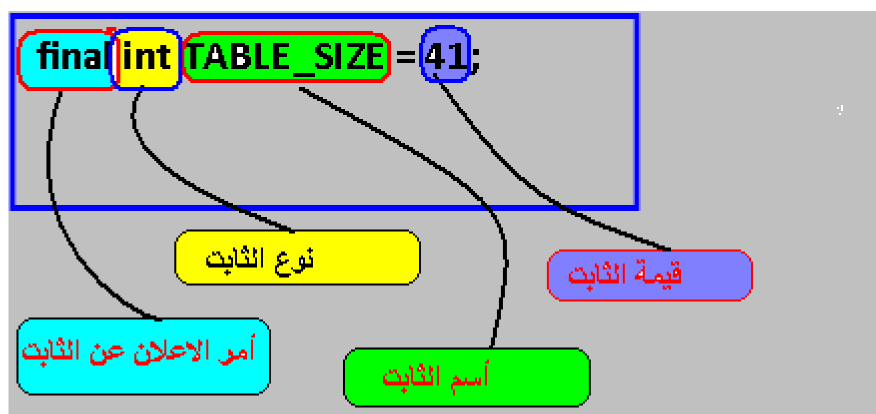
وهى عبارة عن رموز اللغة وتتكون من الحروف والأرقام وتكون بين علامتي تنصيص أو اقتباس ومن الأمثلة على الثوابت الرمزية مثل...(("30+40" - "12345" - "K" - "NAME") وهى بدورها تنقسم الى:

- **ثوابت حرفية..CHARACTER CONSTANTS** وهى التى تتكون رقم او حرف او رمز من متسلسلات الهروب
- **ثوابت السلسلة STRING CONSTANT** يحتوى ثابت السلسلة على اى عدد من الرموز او الفرغات وتكون محصورة بين علامتى تنصيص . " "

<u>كيفية الاعلان عن الثوابت فى الجافا:</u>

للاعلان عن أى قيمة فى تطبيق جافا على أنها ثابت فأننا نستخدم الامر FINAL وذلك لاخبار الحاسب الالى

بحجز مكان فى الذاكرة لحفظ تلك القيمة وانها ستظل ثابتة طوال تنفيذ البرنامج ....ثم بعد نحدد نوع الثابت ثم وضع أسم للثابت



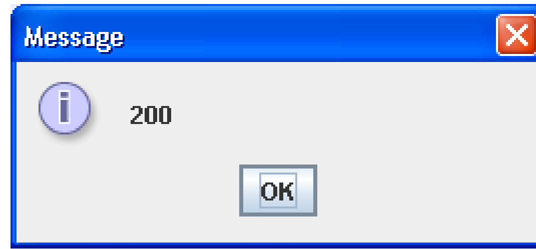
FINAL DATATYPE CONSTANT_NAME =CONSTANT_VALUE

مثال : قم بإظهار رسالة للمستخدم مع تخزين الرقم ٢٠٠ على أنه ثابت .... الحل كما يلى:

FINAL INT EYG=200;

```
1
2   package m;
3   import javax.swing.JOptionPane;
4   public class M {
5
6       public static void main(String[] args) {
7        final int eyg=200;
8        JOptionPane.showMessageDialog(null, eyg);
9       }
10  }
```

ويجب أن تلاحظ أى الكريم أننا أستخدمنا الامر FINAL متبوعة بنوع البيانات التى سأجعلها ثابته وبما أن المثال فرض علينا الاعلان عن ثابت رقمى صحيح لذلك حددنا نوعه INT وذلك لان الرقم الذى سوف نخزنة فى هذا الثابت رقم صحيح ليس به علامة عشرية .أما مخرجات البرنامج ستكون كما يلى:



مثال أخر ....قم بإظهار رسالة للمستخدم مع تخزين الرقم ٥,٥ على أنه ثابت..الحل

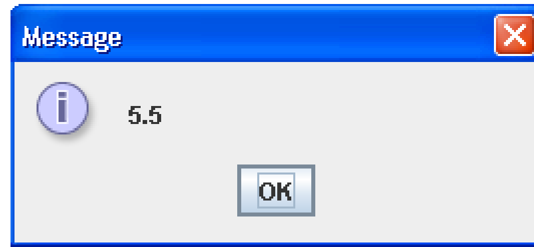
FINAL FLOAT EYG=(FLOAT) 200.25;

```
1
2   package m;
3   import javax.swing.JOptionPane;
4   public class M {
5
6       public static void main(String[] args) {
7
8         float egy=(float)5.5;
9
10
11        JOptionPane.showMessageDialog(null,egy);
12
13      }
14  }
```

ويجب ملاحظة أخى الكريم أننا حددنا نوع الثابت هنا على FLOAT وذلك لانه رقم يحتوى على علامة عشرية وبالتالى فهو من الاعداد الحقيقية كما أننا فى تلك الحالة وضعنا الامر FLOAT بين قوسين عندما أعطينا قيمة للثابت .....كما يمكنك أيضا الاعلان عن الاعداد الحقيقية كما يلى:

```
1
2   package m;
3   import javax.swing.JOptionPane;
4   public class M {
5
6       public static void main(String[] args) {
7        final double eyg=5.5;
8        JOptionPane.showMessageDialog(null, eyg);
9       }
10  }
```

لاحظ أخى الكريم أننا أستخدمنا DOUBLE للاعلان عن الثابت EGY ثم وضعنا قيمته تساوى ٥,٥ ....ومخرجات البرنامج فى كلا الحالتين كما فى الصورة التالية:



ثانيا : المتغيرات:

هى عبارة عن قيم أيضا يتم تخزينها فى ذاكرة الحاسب ولكنها يمكن تغيرها اثناء فترة تشغيل البرنامج وتنقسم الى:

أ ـ متغيرات رمزية (حرفية). ...وهى التى تتضمن الحروف بكافة أشكالها والرموز والفراغات (مسافة فارغة :)

ب ـ متغيرات عددية....وهى التى تتضمن الارقام سواء إن كانت أرقام صحيحة _ الارقام التى بدون علامة عشرية حتى الرقم ٣٢٧٦٧ ....أو كانت أرقام حقيقية وهى جميع الارقام والتى تحتوى على علامة عشرية

كيفية الاعلان عن المتغيرات:

توجد طريقتان للاعلان عن المتغيرات .....

الطريقة الاولى تكون بهذا الشكل:

DATATYPE VARIABLES_NAME;
VARIABLES_NAME=VARIABLES_VALUE ;

Datatype Variables_Name;

نوع المتغير — أسم المتغير

Variables_Name = Variables_Value ;

أسم المتغير — قيمة المتغير

فكما ترون فى تلك الطريقة يتم الاعلان عن المتغير ثم بعد ذلك يتم تخصيص هذا المتغير بوضع قيمة له ....ويمكنك أختصار تلك الطريقة كما يلى:

```
Datatype Variables_Name=Variables_Value ;
```

وهذه هى **الطريقة الثانية** يتم الاعلان و التخصيص فى سطر واحد

أمثلة توضيحية....

```
char a,b;

a= 'a';
b='/ ';
```

فى المثال السابق قمنا بالاعلان على عدد ٢ متغير حرفى ثم بعد ذلك خزنا فى المتغير الاول a القيمة a ويجب ملاحظة أننا وضعنا القيمة بين علامتى إقتباس كما يمكننا استخدام هذا المثال كما فى الشكل التالى:

```
char a="a",b="/";
```

أما الاعلان عن المتغيرات الرقمية الصحيحة تكون بهذا الشكل

```
int a;
A=100;
```

أو

```
int a=100;
```

ويجب ملاحظة أن قيمة المتغير رقم صحيح ١٠٠ أما اذا كان ذو علامة عشرية فإنه بذلك أصبح عدد حقيقى فيكون الاعلان عنه بهذا الشكل:

```
float x;
X=5.2;
```

أو

```
float x= 5.2
```

وهنا يجب علينا الانتباه لجملة الإعلان والتخصيص السابقة , float x= 5.2 إلا أن الحاسب يعتبره من النوع double أي يحجز له مكان ٦٤ بت وللتغلب على المشكلة السابقة يتم الإعلان والتخصيص كالتالي:

```
float x= 5.2 f
```

أي يتم وضع حرف (f) بعد الرقم لكي يتم حجز مكان له في الذاكرة مقداره ٣٢ بت وبذلك نكون قد وفرنا في الذاكرة المستخدمة....أما المتغيرات العددية الحقيقة الطويلة Double هي نفس المتغيرات العددية الحقيقة float ولكن يمكن تمثيلها في خمسة عشرة خانة وتكتب على الشكل التالي:

```
double x;
```

قواعد تسمية الثوابت و المتغيرات فى الجافا:

1_يفضل أن يكتب أسم الثابت بالحروف الكبيرة CAPITAL LETTERS فى حين أن أسماء المتغيرات يفضل أن تكون بالاحرف الصغيرة.. SMALL LETTERS ويجب ملاحظة أخى الكريم أننا نقول يفضل وليس يجب معنى ذلك إن لم تلتزم بهذه القاعدة فإن البرنامج سيعمل معك بشكل صحيح.

2_يمكن أن يبدأ الاسم فى الثوابت و المتغيرات بالحرف أو الشرطة السفلية UNDER SCORE (_) أو علامة الدولار ($) ولكن لا يمكن بدء التسمية برقم ولكن يمكن أن نضع رقم بعد الحرف.
-لا يمكن تسمية المتغيرات أو الثوابت بإحدى الكلمات المحجوزة

العمليات الحسابية و المنطقية

معظم برامج الكمبيوتر تستخدم العمليات الحسابية والشكل التالى يوضح كيفية تمثيل تلك العمليات فى لغة الجافا ولاحظ الفرق بين التعبير فى الجافا و التعبير الجبرى ويمكن تلخيصها فيما يلى:

العملية	العامل	التعبير الجبري	التعبير بالجافا
جمع	+	f+7	f+7
طرح	-	f-7	f-7
ضرب	*	bm	b*m
قسمة	/	x	x/y
موديلاس	%	r mod s	r%s

- علامة **+** والغرض منها عملية الجمع بين العوامل
- علامة **-** والغرض منها عملية الطرح بين العوامل
- علامة ***** والغرض منها ضرب العوامل
- علامة **/** والغرض منها القسمة بين العوامل وعند استخدام هذا المؤثر نحترس حيث يمكن ان يكون العاملين المستخدمين اعداد صحيحة وينتج عن العملية عدد عشرى a=15 , b= مثال ٤ اعداد a,b هذه هى الاجابة المتوقعة ولن الناتج سيكون غير ذلك حيث ان العاملان a/b=3.75 صحيحة فيكون الناتج عدد صحيح فيكون الناتج ٣ فقط دون العدد الكسرى او العشرى ولكن ماذا لو ارادنا القسمة الصحيحة فى هذه الحالة يجب ان يكون احد العاملين عدد عشرى بمعنى وهى نفس القيم القديمة ولكن اختلف نوعها وفى هذه الحالة يكون b= او a= ٤,٠ ان تكون ١٥,٠الناتج ٣,٧٥ لذلك احترس عزيزى القارئ عند استخدامك هذا المؤثر.

- علامة **%** ويكون ناتج هذه العملية هو باقى القسمة ونلاحظ هنا انه يجب ان تكون العوامل اعداد صحيحة لضمان صحة الناتج وتكون اشارة الناتج هى نفس اشارة العامل الايسر...

<u>مثال .....قم بعرض رسالة بها مجموع العددين ٥ و ٦,٥ ......الحل كما يلى:</u>

<u>تكبير الصورة معاينة الأبعاد الأصلية.</u>

```
1
2   package m;
3   import javax.swing.JOptionPane;
4   public class M {
5
6       public static void main(String[]
7
8         int a=5;
9         float b=6.5f;
10        float c=a+b;
11       JOptionPane.showMessageDialog(null,"المجموع هو "+ c);
12
13      }
14  }
```

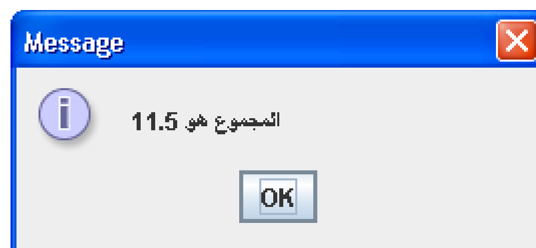
الاعلان عن المتغير الاول

الاعلان عن الرقم الثانى

الاعلان عن المتغير الذى سيتم تخزين نتيجة العملية الحسابية فيه لحين عرضها

تم وضع علامة + وذلك لعرض جملة (المجموع هو)بالاضافة الى نتيجة العملية الحسابية

مخرجات البرنامج كما بالشكل التالى:



عامل الزيادة وعامل النقصان:

تمندنا لغة الجافا بعامل الزيادة ++ ويستخدم لزيادة المتغير أو الثابت بمقادر واحد ...فمثلا أذا كان قيمة المتغير a = 5 فذاذا وضعت عامل الذيادة قبل المتغير هكذا a ++ فمعنى ذلك أن قيمة هذا المتغير أصبحت ٦ ....و عامل النقصان -- ويستخدم لنقص المتغير أو الثابت بمقدار واحد صحيح ففى المثال السابق إذا وضعت عامل النقصان قبل المتغير هكذا -- aفمعنى ذلك أن قيمة هذا المتغير أصبحت ٤...

<u>ويمكن وضع تلك العوامل قبل المتغير او بعده ولكن بالتاكيد هناك فرق بينهم</u>

لاحظ معى اذا كانت a=5 وكانت هناك معادلة بهذا الشكل... b=(a++)*3 فإن الناتج سيكون b=15 ولكن اذا أستخدمنا المتغير a فى عملية أخرى سيكون قيمته ٦ ...أما اذا كانت المعادلة بهذا الشكل b=(++a)*3 فإن الناتج سيكون b=18 هنا الزيادة تمت على المتغير a و اصبح قيمته ٦ قبل العملية الحسابية

اذن اذا كان عامل الزيادة أو النقصان قبل المتغير فتتم الزيادة أو النقص فى نفس السطر او فى نفس الخطوة فى البرنامج لكن اذا كان بعد المتغير فتتم الزيادة بعد الانتهاء من هذه الخطوة...

الفصل الرابع : طرق الادخال و الاخراج فى الجافا

فى هذا الدرس إن شاء الله تعالى سنتعلم سويا ونتعرف على طرق أدخال و أخراج البيانات فى لغة الجافا ...وهذا الدرس مهم جدا حيث لا يخلو أي برنامج من طريقة لادخال البيانات و استخراجها وهي الوسيلة الوحيدة التي يستطيع فيها المستخدم إدخال القيم عن طريق لوحة المفاتيح للحاسب ، حتى يقوم بمعالجة هذه القيم سواء كان البرنامج (برنامج حسابات – قاعدة بيانات. (......

والحقيقة أن لغة الجافا تحتوي على أكثر من طريقة لإدخال البيانات و أخراجها منها ما هو مناسب للتطبيقات الرسومية (تطبيقات الويندوز (ومنها ما هو مناسب التطبيقات غير الرسومية _تطبيقات الدوس (Console Application) و سوف يتم شرح تلك الطرق فى البيئتين.

أولا :طريقة إدخال وإخراج البيانات فى التطبيقات الرسومية:

ويتم ذلك عن طريق اوامر الكلاس المسمى JOptionPane حيث يمدنا بنوافذ حوارية مختلفة تساعدنا على إدخال و اخراج البيانات من والى البرنامج من خلال تلك النوافذ...

1_طريقة إدخال البيانات بأستخدام: JOptionPane

أولا : يجب تحميل الكائن أو الكلاس المسمى JOptionPane من الحزمة المسماة **javax.swing** وهذه الحزمة تحتوى على كثير من الكائنات الاخرى _ستيم شرحها فى حينها _ويتم ذلك عن طريق الكود التالى:

الرمز:

```
import javax.swing.JOptionPane;
```

ثانيا : يجب تعريف المتغيرات التى سيتم تخزين ما يقوم المستخدم بإدخالة عن طريق لوحة المفاتيح ولابد أن تكون من النوع النصى String وذلك لان كل بيانات يتم أدخالها من المستخدم بواسطة هذا الأمر يعتبر.... String

الرمز:

```
String name;
```

ثالثا :نقوم بإظهار المربع الحوارى لادخال البيانات من المستخدم ويتم تخزينها فى المتغير String

الرمز:

```
name=JOptionPane.showInputDialog("   enter name  ");
```

رابعا :إنهاء البرنامج وذلك بإستخدام الامر

الرمز:

```
System.exit( 0 );
```

و كقاعدة عامة يجب أستخدام هذا الامر لانهاء التطبيق فى التطبيقات الرسومية ....وذلك لان التطبيق فى حالة أستخدام هذا الامر سيكون نشط فى ذاكرة الجهاز.....

اذن الكود العام لادخال البيانات بأستخدام الكلاس JOptionPane سيكون كما فى الشكل التالى:

```
1
2    package m;
3    import javax.swing.JOptionPane;
4    public class M {
5
6        public static void main(String[] args) {
7
8            String name;
9
10           name=JOptionPane.showInputDialog( ("من فضلك أدخل أسمك")
11           System.exit( 0 );
12       }
13   }
```

الآن انتهينا من كيفية ادخال البيانات بتلك الطريق لكن ماذا لو كنا نريد من المستخدم إدخال بيانات رقمية و ليست نصية و نريد أستخدمها كأرقام فى عمليات حسابية
فقد أخبرنا سابقا بأن كل شيء يتم أخذه من المستخدم بواسطة هذا الأمر يعتبر) String سيتم شرحها لاحقاً بالتفصيل) ، بمعنى أن البرنامج سيعتبرها كلمات أو جمل أو نصوص، ولذلك لن يتمكن من التعامل معها في المعادلات الرياضية والحسابية حتى لو أدخل المستخدم فيها أرقام.بمعنى لو أدخل المستخدم رقم سيتم تخزينه في البرنامج بين علامتي التنصيص بهذا الشكل: "٩٩"

ووجوده بين علامتي التنصيص يعني انه لايعتبر من أنواع المتغيرات التي يمكن التعامل معها في المعادلات الرياضية و الحسابية مثل int و double و byte وغيرها.......
ولو لم نقم بعملية تحويلها إلى بيانات رقمية ، وحاولنا أن نقوم بعمليات حسابية لها مباشرة ، سيحدث خطأ في البرنامج ، لأنه لن يستطيع التعامل مع الـ String في العمليات الحسابية.

لذلك توجد أوامر لتحويل المدخلات من بيانات نصية إلى أنواع البيانات الرقمية التي نريدها....وذلك على النحو التالى:

```
byte b = Byte.parseByte("123");
short s = Short.parseShort("123");
int i = Integer.parseInt("123");
long l = Long.parseLong("123");
float f = Float.parseFloat("123.4");
double d = Double.parseDouble("123.4e10");
```

كيفية تحويل النصوص المدخلة الى أنوع من متغيرات رقمية

و هذا الشكل يوضح الاوامر المستخدمة لتحويل القيم النصية المدخلة بواسطة المستخدم الى قيم رقمية و ذلك لاستخدامها فى العمليات الحسابية.....

مثال : بقم بعرض رسالة للمستخدم لاخذ بيانات نصية........**الحل كما يلى:**

```java
package m;
import javax.swing.JOptionPane;
public class M {

    public static void main(String[] args) {

      String no;

     no=JOptionPane.showInputDialog("من فضلك أدخل الرقم")
     System.exit( 0 );
    }
}
```

ستجد أخى الكريم أن هذا المثال هو نفس المثال السابق بدون أى تغير ....وفى تلك الحالة أى نص أو حرف أو حتى أرقام أدخلها المستخدم فأنها سوف يتم حفظها فى المتغير **(no)** على أنها بيانات نصية و ليست رقمية ....بمعنى لو أدخل المستخدم الرقم (5) ستعبر البرنامج أن تلك القيمة نصية و لن تستطيع أستخدمها فى العمليات الحسابية وسيحدث خطأ للبرنامج لو أستخدمتها......

مثال أخر ....قم بعرض رسالة للمستخدم لاخذ بيانات رقمية..... **الحل كما يلى:**

```java
package m;
import javax.swing.JOptionPane;
public class M {

    public static void main(String[] args) {

      String no;
     int number;
     no=JOptionPane.showInputDialog("من فضلك أدخل الرقم")
     number=Integer.parseInt("no");
     System.exit( 0 );
    }
}
```

لاحظ أى الكريم ...التغير الذى حدث فى المثال الثانى عن المثال الاول ...فقط قمنا بإضافة سطرين جديدين ...الاسطر الاول هو الاعلان عن المتغير number على أنه من نوع بيانات رقمية صحيحة... int أما السطر الاخر هو أننا قمنا بتحويل القيمة النصية المخزنة فى المتغير no الى بيانات رقمية صحيحة من نوع int و تم تخزينها فى المتغير number ....وهكذا نكون قد حولنا القيمة البيانات النصية من المستخدم الى بيانات رقمية و تستطيع الان أجراء العمليات الحسابية عليها....

كما يرجى ملاحظة اننا حولنا القيم النصية الى قيم رقمية من نوع... int أما اذا كان نريد أنواع رقمية أخرى غير ذلك فاذا كانت البيانات المتوقعة ادخالها من المستخدم من نوع byte اذن كود التحويل سيكون كما يلي:

الرمز:

```
number = Byte.parseByte("no");
```

واذا كانت البيانات المتوقعة أدخلها من المستخدم من نوع long اذن كود التحويل سيكون كما يلى :

الرمز:

```
number= Long.parseLong("no");
```

واذا كانت البيانات المتوقع أدخالها من نوع Float اذن كود التحويل سيكون كما يلى:

الرمز:

```
number = Float.parseFloat("no");
```

وهكذا بالنسبة للبيانات من نوع **double** أو **short وبما أن لغة الجافا لحالة الاحرف فهناك خطأ شائع وهو عدم الالتزام بنفس الجمل السابقة من حيث الاحرف الكبيرة و الصغيرة.**

2_طريقة أخراج البيانات بأستخدام: JOptionPane

طريقة اخراج البيانات بأستخدام الكلاس المسمى JOptionPane كنا قد تعودنا على أستخدمها فى الدروس السابقة ...وقلنا أننا اذا أردنا أن نخرج بيانات للمستخدم على هيئة نافذة رسالة يكون عن طريق جملة الاخراج التالى:

الرمز:

```
JOptionPane.showMessageDialog(null, "");
```

وكانت هذه ابسط رسالة يمكن أظهارها للمستخدم و تحتوى بيانات ...لكن يمكنا اظهارها بصوة أكثر تنسيق باستخدام جملة الاخراج التالية:

```
showMessageDialog(null, message, title, messageType)
```

وجملة الاخراج السابقة فكما هو واضح تتطلب أربع مدخلات:

- المدخل الاول : الكلمة المحجوزة null وفائدتها وضع صندوق الحوار فى وسط الشاشة.
- المدخل الثانى message: نص الرسالة الذى سوف يظهر للمستخدم ويجب وضع علامتى أقتباس على النص المراد أخراجة للمستخدم
- المدخل الثالث title: عنوان الرسالة الذى سوف يظهر فى سطر العنوان للرسالة
- المدخل الرابع messageType: وهو نوع الرسالة حيث يوجد مجموعة من الرموز التى التى يمكن أظهارها فى صندوق الحوار التى تساعد المستخدم فى معرفة نوع صندوق الحوار و الرسالة التى تظهر فية ....

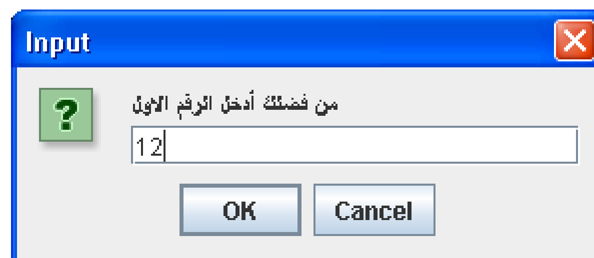
❖ والشكل التالى يوضح تلك الرموز:

نوع رسالة صندوق الحوار	الرمز	الوصف
JOptionPane. ERROR_ MESSAGE		عرض صندوق حوار يبين رسالة خطأ للمستخدم
JOptionPane. INFORMATION_ MESSAGE		عرض صندوق حوار مع رسالة للمستخدم
JOptionPane. WARNING_ MESSAGE		رسالة تحذيرية للمستخدم
JOptionPane. QUESTION_ MESSAGE	?	سؤال للمستخدم يجب الإجابة عليه بنعم أو لا
JOptionPane. PLAIN_ MESSAGE	لا يوجد رمز	يظهر رسالة في الصندوق بدون رموز

مثال : قم بعمل برنامج يقوم بحساب عددين يدخلهم المستخدم و إظهار الناتج فى مربع حوارى..........**الحل:**

```
1  package m;
2  import javax.swing.JOptionPane;
3  public class M {
4
5      public static void main(String[] args) {
6
7        String no1,no2;
8        int number1,number2;
9        long total;
10
11       no1=JOptionPane.showInputDialog("من فضلك أدخل الرقم الاول");
12       no2=JOptionPane.showInputDialog("من فضلك أدخل الرقم الثانى");
13
14       number1=Integer.parseInt(no1);
15       number2=Integer.parseInt(no2);
16
17       total=number1+number2;
18      JOptionPane.showMessageDialog(null,"الناتج هو" + total, "egy_tech", 2);
19
20      System.exit( 0 );
21     }
22 }
```

مخرجات البرنامج:





شرح الكود المستخدم فى البرنامج:

السطر رقم ٧ : قمنا بتعريف متغريين no1 ,no2 من نوع نصى String وذلك لتخزين القيم التى يتم أخذها من المستخدم ...وطبعا قلنا أن تلك القيم يعتبرها البرنامج من النوع النصى حتى و أن كانت أرقام

السطر رقم : 8_قمنا بتعريف متغريين number1,number2 من النوع int وذلك لتخزين القيم المدخلة من المستخدم فيهما بعد تحويلها الى قيم رقمية من نوع int.

السطر رقم ٩ : قمنا بالاعلان عن المتغير total وذلك لتخزين ناتج العملية الحسابية فيه..

السطر العاشر و الحادى عشر : قمنا باظهار مربعين أو رسالتين أدخل للمستخدم عن طريق الامر showInputDialogللكلاس المسمى JOptionPane وذلك لاخذ القيم من المستخدم و تخزينها فى المتغريين no1 ,no2على التوالى.

السطر الرابع عشر و الخامس عشر : قمنا بتحويل القيم المخزنة فى المتغريين no1 ,no2 الى قيم رقمية من نوع int عن طريق الامر... Integer.parseInt وتخزين ناتج التحويل فى المتغريين number1,number2و الذى قمنا بالاعلان عنهم سابقا...

السطر السابع عشر : جعلنا المتغير total يساوى قيمة المتغير number1 بالاضافة الى قيمة المتغير number2

السطر الثامن عشر : أظهرنا مخرجات البرنامج عن طريق مربع حوارى بأستخدام الامر showMessageDialogللكلاس المسمى... JOptionPane ويرجى ملاحظة أننا أستخدمنا المدخلات الاربعة سالفة الذكر بين قوسين فى إظهار المربع الحوارى ....فأستخدمنا المدخل (null) لاظهار المربع الحوارى فى منتصف الشاشة ....و مدخل نص الرسالة وفى هذا المثال هو" الناتج هو (total + "وهنا تم أستخدام العامل (+) وذلك لالحاق القيمة المخزنة فى المتغير total بعد النص "الناتج هو " ......ثم بعد ذلك مدخل عنوان الرسالة و هو فى مثالنا هذا......" egy_tech" أما الرقم ٢ الموجود فى الكود هو المدخل الرابع الذى يحدد نوع ايقونة الرسالة وقد تم تغيرها فى مثالنا هذا كما هو ظاهر فى مخرجات البرنامج,,,

ثانيا :طرق الادخال و الاخراج فى البيئة غير الرسومية,,,

1_طريقة أخراج البيانات فى البيئة غير الرسومية:

الطريقة التى تستخدم لاخراج البيانات فى البيئية غير الرسومية تسمى Standard output ويتم ذلك عن طريق أستخدام الكلاس.... System و ذلك بأستخدام الكود التالى:

الرمز:

```
System.out.print("Egy_tech!");
```

ويجب ملاحظة أن حرف S كابيتال فى كلمة... System والنص المطلوب عرضة Egy_tech فى قوسين وبين علامتى إقتباس ...ونلاحظ أنه عند تنفيذ هذا البرنامج ستظهر العبارة (Egy_tech) على سطر واحد ....أما إذا أردنا أن تكون المخرجات على سطرين مختلفين فيتم إضافة حرفي (**ln**) على العبارة print ومعناها الانتقال إلى سطر جديد (new line)ويتم ذلك في السطر الثالث كالآتي:

الرمز:

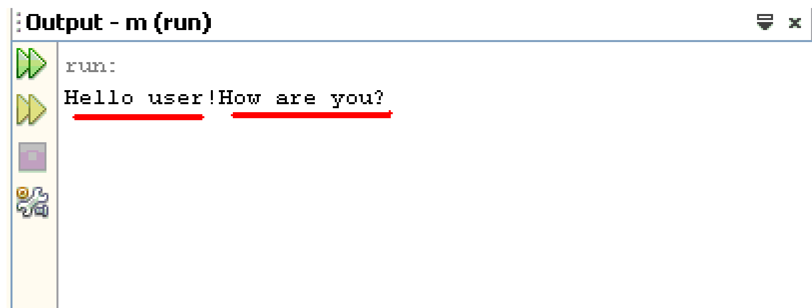
```
System.out.println("Egy_tech");
```

الفرق بين الأول والثاني ، هو أن الثاني سيجعل أمر الكتابة الذي بعده يكون في سطر جديد ، ولكن الأول لن يقوم بهذا. مثلاً لو كتبنا هذه الأمرين:

الرمز:

```
System.out.print("Hello user!");
System.out.print("How are you?");
```

ستكون النتيجة كذلك



فكما تلاحظ أخى الكريم أن الجملتيين فى سطر واحد .... أما اذا استخدمنا جملة الاخراج كما يلى :

الرمز:

```
System.out.println("Hello user!");
System.out.println("How are you?");
```

ستكون النتيجة كما يلى:

```
Output - m (run)
run:
Hello user!
How are you?
```

ويجب ملاحظة الفرق هنا ...فقد تم كتابة الجملتين كل جملة فى سطر ....وهذا هو الفرق بين print و println فى إخراج البيانات ....ويمكن تنفيذ نفس شكل المخرجات السابقة بسطر واحد وذلك عن طريق إضافة (n\) وتعني الانتقال إلى سطر جديد ويكون شكل البرنامج كالتالي:

الرمز:

```
System.out.print("Hello user!\n How are you?");
```

فتكون المخرجات نفس المخرجات السابقة بهذا الشكل

```
Output - m (run)
run:
Hello user!
How are you?
```

أما اذا أردنا طباعة عدة أسطر متتالية بأمر طباعة واحد فيتم ذلك بتكرار (n\) كالتالي:

الرمز:

```
System.out.print ("One\n Two \n Three \n Four") ;
```

فتكون المخرجات كالتالي

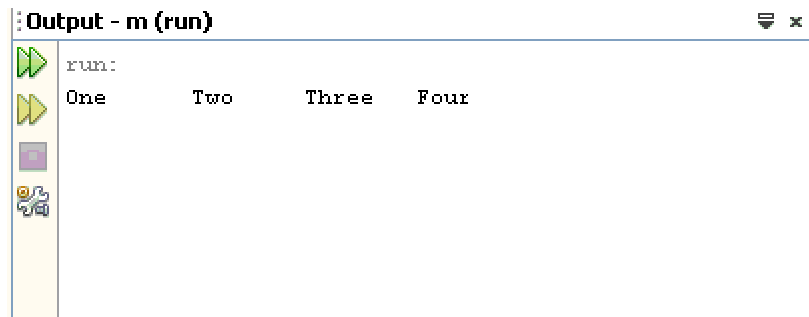
```
Output - m (run)
run:
One
 Two
 Three
 Four.
```

أما اذا أردنا أن تكون المخرجات على مسافات أفقية متساوية فأننا نستخدم (t\) كالتالي:

الرمز:

```
System.out.print ("One\t Two \t Three \t Four") ;
```

وتكون المخرجات كالتالي:



وكل هذه الاوامر تسمى **متسلسلات الهروب** والتي يتم إضافتها لجملة الطباعة للحصول على عدة اشكال للمخرجات ....واليكم الجدول التالى الذى يوضح متسلسلات أخرى:

الحرف الخاص	الوصف
\n	سطر جديد. يضع المؤشر في بداية السطر التالي
\t	مسافة أفقية. تحريك المؤشر مسافة معينة إلى النقطة التالية في السطر
\r	carriage return. يضع المؤشر في بداية السطر الحالي ولا يتقدم إلى السطر التالي ، وأي حرف يطبع يتم طباعته على حرف سابق تم كتابته في نفس السطر
\\	شرطة خلفية. إظهار "\" في الخرج
\"	علامة تنصيص مزدوجة. إظهار علامة التنصيص المزدوجة

2_طريقة إدخال البيانات فى البيئة غير الرسومية:

من الطرق التى تستخدم فى إدخال البيانات فى لغة الجافا هي أستخدام الكلاس(Java.util.Scanner) ، وهي قوية جداً حيث أنه يمكنها ليس فقط قراءة أسطر الإدخال التي يقوم المستخدم بكتابتها، بل يمكنها أيضاً قراءة محتوى الملفات...الكلاس Scannerوكما يدل على ذلك اسمها، فيمكن أن نترجمها بالعربي إلى الماسح، وكلنا نعرف أداة الماسح الضوئي التي تقوم بمسح وثيقة معينة وتحويلها لوثيقة رقمية. هذه الكلاس تقوم بنفس العمل ! فهي تقوم بمسح كل ما تمت كتابته من طرف المستخدم ليتم تخزينه في متغير للإشتغال به في البرنامج ....الطريقة هي سهلة الاستخدام وذلك لأن جميع وظائفها معرفة من طرف الجافا مسبقاً يكفي فقط تتبع الخطوات البسيطة المشروحة هنا.

<u>أولاً: يجب إستدعاء الكلاس Scanner</u> وهي الكلاس التي تحتوي على وظائف القراءة من سطر الإدخال من مكتبة الجافا بالأمر التالي:

الرمز:

```
import java.util.Scanner;
```

ثانيا :نقوم بتعريف كائن له نفس خصائص الكلاس سكانر ليتم تخزين فيه ما قام المستخدم بكتابته عن طريق لوحة المفاتيح و القاعدة العامة لتعريف هذا الكائن تكون كما فى الشكل التالى:

Scanner variable = new Scanner(System.in) ;

اي اسم متغير
(a , b , sum ,.....)

فمثلا أذا أردت أن أنشيء الكائن input له نفس خصائص الكلاس سكانر يكون الامر على النحو التالى:

الرمز:

```
Scanner input = new Scanner(System.in);
```

ثالثا :نقوم بقراءة ما أدخلة المستخدم على حسب ما نود تخزينة من بيانات ...فمثلا إذا كنا نريد قراءة رقم صحيح من نوع...int فإننا سنستخدم الكود التالى للقراءة:

الرمز:

```
input.nextInt();
```

أما إن كانت نوع البيانات غير ذلك .....وكما نعلم هناك عدة أنواع من الأعداد (أعداد طبيعية، عشرية ...) ولكل نوع هناك نوع بيانات معين) كما رأينا في الدرس السابق : أنواع البيانات والمتغيرات .(هذه الأعداد، كما يتم تخزينها في نوع معين من أنواع البيانات، يتم أيضاً قراءتها بنوع معين من أنواع البيانات... فالأعداد الطبيعية يتم قراءتها بواسطة السطور التالية:

1.nextInt
2.nextByte
3.nextShort
4.nextLong

أماالأعداد العشرية يمكن قراءتها بالسطور التالية :

1.nextFloat
2.nextDouble

أما قراءة الكلمات تكون بهذا الامر nextLine

رابعا :يفضل دائماً أن نقوم بإنهاء هذا النوع من الكلاس عندما ننتهي من استعمالها بالسطر التالي:

الرمز:

```
input.close();
```

يجب مراعاة كتابة Scanner بحرف S كبير وكذلك كتابة nextInt ب I كبيرة مع عدم نسيان الفاصلة المنقوطة في سطر تعريف الكائن وكذلك في سطر القراءة بواسطةnextInt

مثال توضيحيى :أكتب برنامج لجمع رقمين على أن يتم إدخال الرقمين من لوحة المفاتيح ثم طباعة الناتج على الشاشة.

الحل :

```
1 import java.util.Scanner  ;   ← استدعاء المكتبة التي تحتوى على جملة الأدخال
2  public class magdy{
3 public static void main(String s[] ){
4 int  a,b,sum;   ← الأعلان عن ثلاث متغيرات
5 Scanner Keyboard=new Scanner(System.in);   ← تعريف كائن عملية الادخال
   6 System.out.println("Enter first number");
   7 a=Keyboard.nextInt();   ← هنا يتم قراءة الرقم الأول وتخزينه في المتغير a
   8 System.out.println("enter seconde number");
   9 b=Keyboard.nextInt();   ← هنا يتم قراءة الرقم الثانى وتخزينه في المتغير b
  10 sum=a+b;
  11 System.out.println("the sum is="+sum);
    }
 }
```

الشرح:

- في السطر الأول تم استدعاء المكتبة التي تحتوي على جملة الإدخال Scanner

```
1 import java.util.Scanner  ;
```

- أما في السطر الرابع فقد تم الإعلان عن ثلاث متغيرات من النوع integer وهي المتغير a لتخزين الرقم الأول والمتغير bلتخزين الرقم الثاني والمتغير sum لتخزين ناتج عملية الجمع .

```
4  int  a,b,sum;
```

- أما السطر الخامس تعريف كائن له نفس خصائص الكلاس سكانر لاستقبال مدخلات من لوحة المفاتيح ولقد تم تسميته Keyboardو يمكن تسميته بأي اسم تريدة ويتم فيه تخزين القيمة المدخلة مؤقتا تمهيدا لنقلها لمتغير أخر سيكون هنا a أو b ولاحظ كلمة (System.in) أصبح بجوارها كلمة in دلالة على عملية الإدخال

```
5  Scanner Keyboard=new Scanner(System.in);
```

- أما السطر السادس فوظيفته هي طبع رسالة على الشاشة تخبر المستخدم بإدخال الرقم الأول.
-

❖ أما في السطر السابع فيتم قراءة ما تم إدخالة في المخزن Keyboard و وضعها في المخزن a.

```
7  a=Keyboard.nextInt();
```

❖ أما في السطر الثامن فهو يكرر العملية لطلب الرقم الثاني.
❖ وفي السطر التاسع يتم إدخال الرقم الثاني كما سبق ولكن يتم تخزينه هذه المرة في المخزن.b
❖ أما السطر العاشر فيتم فيه عملية الجمع ووضع الناتج في المخزن.sum
❖ وفي السطر الحادي عشر يتم طباعة قيمة الجمع على الشاشة.

أما عيب هذه الكلاس (Scanner) أنها لا تحتوي على وظيفة لقراءة حرف واحدا!............. لذلك ولقراءة حرف واحد يجب قراءة السطر بأكمله ثم بعد ذلك قراءة الحرف الأول من الكلمة أو الجملة التي قام المستخدم بادخالها. وذلك يتم كما في المثال التالي:

```
01. // هنا يتم إستدعاء الكلس سكانر
02. import java.util.Scanner;
03.
04. public class Input_char {
05.
06.     /**
07.     * @author Ayoub El Hobbi
08.     */
09.    public static void main(String[] args) {
10.
11.        // نقوم بتعريف كائن إسمه إنبوت له نفس خصائص الكلاس سكانر
12.        Scanner input = new Scanner(System.in);
13.
14.        // نطلب من المستخدم إدخال الحرف الأول من إسمه
15.        System.out.print("what is the first character of your name ? : ");
16.
17.        // نقوم بتخزين ما قام بكتابته في متغير تاج
18.        String name = input.nextLine();
19.
20.        // نأخذ الحرف الأول مما أدخله المستخدم .. فحتى وإن أدخل إسمه بأكمله
21.        char firstChar = name.charAt(0);
22.
23.        // وهنا تتم طباعته
24.        System.out.println("the letter is :  " +firstChar);
25.        input.close();
26.    }
27.
28. }
```

كما رأينا في هذا المثال فحتى وإن قام المستخدم بكتابة إسمه بأكمله يمكننا تخزنن فقط الحرف الأول من الكلمة، وكما يمكن تخزين موضع الحرف الأول من الكلمة، يمكن أيضاً تخزين الموضيع التاني، الثالث ... وذلك بتعويض رقم ٠ في charAtبمثلاً رقم ١.....لنأخذ مثلاً إسم .Ayoub أول حرف من هذا الإسم هو a وهو يحمل الرقم ٠ فيcharAt ، ثاني حرف يحمل الرقم ١، وهكذا ... يمكنك تجربة ذلك بنفسك

الفصل الخامس : أدوات التحكم البنائى

أهلا وسهلا أخوانى الكرام فى الدرس الخامس من دروس منتدى مصر التقنى لتعلم الجافا الرسومية بأستخدام NetBeans ... درسنا اليوم هام جدا فلا يكاد يخلوا أى برنامج من أحد أدوات التحكم البنائى مثل الجمل الشرطية و حلقات التكرار....

أولا : الجمل الشرطية:

ويقصد بها تنفيذ أمر ما عند تحقق شرط معين ...فلنتصوّر مثلا أننا نكتب برنامجاً صغيراً جداً. يخزّن رقماً في متغير، ثم يقارن هذا المتغير بالرقم عشرة، فإذا كان المتغير أصغر نعرض على الشاشة جملة معينة. و إذا كان أقل من عشرة، سنعرض على الشاشة جملة أخرى.....

1_جملة IF وتنقسم الى عدة صيغ وذلك على النحو التالى:

- **الصيغة الأولى لجملة if** و تأخذ الصيغة العامة التالية:

```
الشرط
if (expression) {
        statement1; // جملة1
        statment2;  // جملة2
    }
```

كما هو موجود فى الشكل السابق نبدأ جملة السرط بالكلمة (if) و بعدها نفتح قوسين دائريين نضع في داخلها شرطاً ما.
بعدها نفتح قوس مجموعة نضع في داخله ما نريد تنفيذه في حالة تحقق الشرط فقم بتنفيذ الجمل التي بين القوسين .....
وفي حالة عدم تحقق فلا يتم تنفيذ جملة (if) وإنما استمر في تنفيذ بقية جمل البرنامج من بعد تخطي جملة(if)

ويجب ملاحظة أن تلك الصيغة أحتوت على عدة سطور برمجية يتم تنفيذها فى حالة تحقق الشرط فتم نفتح قوسين دائريين ....أما في حالة تنفيذ جملة واحدة فقط بعد جملة (if) فإنه يمكن الاستغناء عن الأقواس وفي هذه الحالة تنتهي جملة الشرط بالفاصلة المنقوطة (؛) كما يلي:

```
الشرط
if (expression)
        statement1;
    تنتهي جملة الشرط عند الفاصلة المنقوطة
```

و كما هو معلوم فإن الشرط يجب أن يحتوى على أحد المعاملات المنطقية التالية:

المعنى	المؤشر
يساوى	==
يخالف	!=
اصغر او يساوى	<=
اكبر او يساوى	>=
اكبر قطعا	>
اصغر قطعا	<

مثال للتوضيح : المطلوب كتابة برنامج يقوم بفحص رقم يتم إدخاله من لوحة المفاتيح وليكن (x) بحيث إذا كان أكبر من الصفر يقوم بطباعة العبارة (الرقم موجب)

الحل.....

```
1  package m;
2  import javax.swing.JOptionPane;
3  public class M {
4
5      public static void main(String[] args) {
6
7    String no;
8    int nomber;
9    no=JOptionPane.showInputDialog(null, "من فضلك أدخل الرقم", "egy_tech", 2);
10   nomber=Integer.parseInt(no);
11   if (nomber>0)
12    JOptionPane.showMessageDialog(null, "الرقم موجب ", "Egy_Tech", 2);
13   System.exit( 0 );
14     }
15 }
```

شرح الكود:

- السطر رقم ٧ : قمنا بالاعلان عن المتغير no لتخزين ما يتم إدخال من المستخدم فيه ...وطبعا كما قلنا فى دروس سابقا أنه من نوع نصى string
- السطر رقم ٨ : قمنا بالاعلان عن متغير رقمى من نوع int يسمى nomber و ذلك لتخزين القيمة الموجودة فى المتغير noبعد تحويلها الى قيمة رقمية.
- السطر رقم ٩ : قمنا بإظهار نافذ للمستخدم لاخذ القيم منه و تخزينها فى المتغير. no
- السطر رقم ١٠: تحويل القيمة النصية المخزنة فى المتغير no الى قيم رقمية و تخزينها فى المتغير. nomber
- السطر رقم ١١ : أستخدمنا جملة الشرط if للتأكد من أن القيمة المدخلة من المستخدم أكبر من الصفر...
- السطر رقم ١٢ : اذا كانت جملة الشرط صحيحة_بمعنى أن الرقم المدخل من قبل المستخدم أكبر من الصفر _يتم أظهار رسالة للمستخدم نصها" الرقم موجب"

وفى هذا المثال يجب ملاحظة أننا لم نستخدم الاقواس فى جملة... if أما اذا كانت جملة if تحتوى على أكثر من سطر برمجى يجب وضع الاقواس....

مثال للتوضيح : كرر المثال السابق مع أظهار عدد ٢ رسالة الاولى لاظهار الرقم المدخل من المستخدم ....و الاخرى لعرض نوعة موجب أو سالب.

الحل:

```
1   package m;
2   import javax.swing.JOptionPane;
3   public class M {
4
5       public static void main(String[] args) {
6
7     String no;
8     int nomber;
9     no=JOptionPane.showInputDialog(null, "من فضلك أدخل الرقم", "egy_tech", 2);
10    nomber=Integer.parseInt(no);
11    if (nomber>0) {
12        JOptionPane.showMessageDialog(null, "الرقم المدخل هو " + nomber, "Egy_Tech", 2);
13        JOptionPane.showMessageDialog(null, "الرقم موجب ", "Egy_Tech", 2);
14    }
15
16
17      }
18  }
```

نلاحظ أن جملة الشرط احتوت على أكثر من سطر ولذلك وضعنا الأقواس

- **الصيغة الثانية لجملة if** و تأخذ الصيغة العامة التالية:



وكما تلاحظ من الشكل السابق أخى الكريم ...جملة **if...else** عبارة عن جملة إختيار تمكنك من القيام بعمل ما إذا تحقق شرط معين، و القيام بعمل أخر إذا لم يتحقق الشرط.

في مثالنا السابق، يمكننا أن نوسع الفكرة بحيث يقوم البرنامج بعرض رسالة بأن الرقم موجب إذا كان أكبر من الصفر و إظهار رسالة بأن الرقم سالب إن كان غير ذلك. كما يلى:

```
1   package m;
2   import javax.swing.JOptionPane;
3   public class M {
4
5       public static void main(String[] args) {
6
7     String no;
8     int nomber;
9     no=JOptionPane.showInputDialog(null, "من فضلك أدخل الرقم", "egy_tech", 2);
10    nomber=Integer.parseInt(no);
11    if (nomber>0)
12        JOptionPane.showMessageDialog(null, "الرقم موجب ", "Egy_Tech", 2);
13    else
14        JOptionPane.showMessageDialog(null, "الرقم سالب ", "Egy_Tech", 2);
15
16      }
17  }
```

و يمكننا في البرنامج السابق حذف الأقواس الموجودة بعد جملة if , وكذلك حذف الأقواس الموجودة بعد else وذلك لوجود جملة طباعة واحدة بعد كلاً منهما.

- **الصيغة الثالثة لجملة IF** و تأخذ الشكل التالي :

```
if(expression 1){
  //اذا كان الشرط الاول صحيح يتم تنفيذة الكود الذى هنا
}else if(expression 2){
  //اذا كان الشرط الثانى صحيح يتم تنفيذة الكود الذى هنا
}else if(expression ){
  //اذا كان الشرط الثالث صحيح يتم تنفيذة الكود الذى هنا
}else {
  //يتم تنفيذة الجملة التى هنا فى حالة عدم صحة الشروط السابقة
}
```

وكما ترى من الصيغة السابقة فان جملة if...else if...else يتسخدم فى التأكد من شروط عدة فإذا كان الشرط الاول صحيح يتم تنفيذ الشفرة التى بين القوسين ثم نستخدم التعليمة else if للتأكد من صحة الشرط الثانى فان كان صحيح يتم تنفيذة و هكذا .....أما اذا كانت كل الشروط غير صحيحة أو لم تتحقق فأننا نستخدم التعليمة else.لتنفيذ كود معين

مثال للتوضيح :

```
public class Test {

   public static void main(String args[]){
      int x = 30;

      if( x == 10 ){
         System.out.print("Value of X is 10");
      }else if( x == 20 ){
         System.out.print("Value of X is 20");
      }else if( x == 30 ){
         System.out.print("Value of X is 30");
      }else{
         System.out.print("This is else statement");
      }
   }
```

مخرجات المثال السابق سيظهر رسالة Value of X is 30 ذلك لان البرنامج سيقوم بالمرور على كافة جملة if...else للتأكد من أن المتغير X= 30 و قد تم التحقق منه فى السطر الثالث

- **الصيغة الرابعة لجملة IF** و هى عبارة جمل متداخلة و تأخذ الشكل التالى :

```
if(Boolean_expression 1){
   //تنفيذ شفرة برمجية
   if(Boolean_expression 2){
      // تنفيذ شفرة برمجية
   }
}
```

وكما ترى فى الصيغة السابقة فهى عبارة عن جمل IF متداخل و تسمح لك بتنفيذ أكثر من جملة برمجية فى وجود تحقق عدة شروط

مثال للتوضيح :

```
public class Test {

   public static void main(String args[]){

      int x = 30;
      int y = 10;

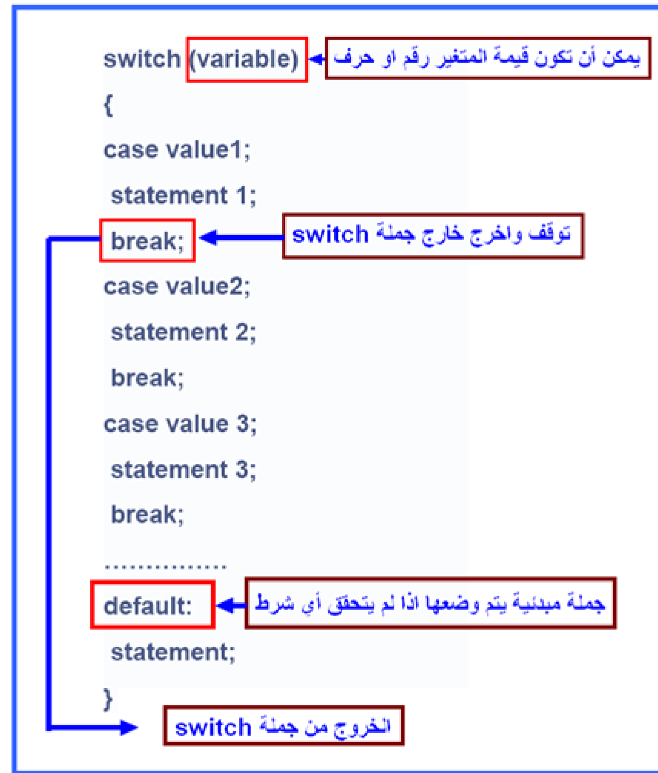
      if( x == 30 ){

         if( y == 10 ){

             System.out.print("X = 30 and Y = 10");
         }
      }
   }
}
```

2_جملة switch

تستخدم عبارة if إذا كان جواب الشرط عبارة عن احتمالين أو ثلاثة احتمالات على الأكثر، أما إذا زاد عدد الاحتمالات على ذلك فمن الأفضل استخدام عبارة switch وصيغتها العامة كالآتي:



ولكي نفهم كيفية عمل جملة switch نجري المثال التالي:

مثال للتوضيح : لنفرض أننا نريد إن يطبع الحاسب جملة ترحيب معينة إذا تم الضغط على أحد الأرقام في لوحة المفاتيح.

الحل:

```
package m;
import javax.swing.JOptionPane;
public class M {

    public static void main(String[] args) {

  String no;
  int nomber;
  no=JOptionPane.showInputDialog(null, "من فضلك أدخل الرقم", "egy_tech", 2);
  nomber=Integer.parseInt(no);
  switch (nomber) {
      case 1:
          JOptionPane.showMessageDialog(null, "السلام عليكم ", "Egy_Tech", 2);
          break;
    case 2:
          JOptionPane.showMessageDialog(null, "صباح الخير ", "Egy_Tech", 2);
          break;
    case 3:
          JOptionPane.showMessageDialog(null, "مساء الخير  ", "Egy_Tech", 2);
          break;
   default:
      JOptionPane.showMessageDialog(null, " الرقم الذى أدخلته غير موجود لدى البرنامج ", "Egy_Tech", 2);
         }
    }
}
```

وعند تنفيذ هذا البرنامج فأنه تحدث إحدى الحالات الآتية:



- ❖ إذا تم إدخال الرقم 1 فإنه يطبع العبارة الأولى "السلام عليكم" ثم يجد عبارة break فيخرج خارج جملة switch وينتهي البرنامج.
- ❖ إذا تم إدخال الرقم 2 فإنه يطبع العبارة الثانية "صباح الخير" ثم يجد عبارة التوقف break فيخرج خارج جملة switch وينتهي البرنامج.
- ❖ وهكذا في حالة إدخال الرقم (٣) فإنه يطبع الجملة الثالثة ثم break ثم يخرج.
- ❖ أما في حالة إدخال أي رقم غير موجود في البرنامج وليكن (٤) مثلا، فإن البرنامج يطبع العبارة الموجودة في جملة "defaultعفوا الرقم غير موجود ...." ثم ينتهي البرنامج.

ملحوظة:

_يجب مراعاة أن جملة switch لها قوسي بداية ونهاية.....و يجب دائماً أن تنتهي كل حالة case من حالاتswitch بالعبارة break حاول أن تقوم بإلغاء هذه العبارة وتجربة البرنامج ... ماذا تجد.؟؟ جرب إدخال حرف بدلا من الرقم ماذا تجد ؟؟

ثانيا : حلقات التكرار

تستخدم الحلقات التكرارية او جمل الدوران فى كل لغات الحاسوب لتساهم فى أداء العمليات التى تتكرر فى البرنامج ... فى عبارة عن بناء برمجي يحتوي علي جمل برمجية قابلة للتنفيذ مرارا تحت تأثير شرط معين ..بمعنى أنه طالما الشرط محقق يتم تنفيذ الكود البرمجى الذى بداخل تلك الحلقة وتحتوى لغة الجافا على ما يلي من الحلقات التكرارية:

- while
- do...while
- for

- الحلقة التكرارية while

و تستخدم تلك الحلقة فى تكرار تنفيذ كود معين لعدد معين من المرات و يتم بنائها على النحو التالى :

```
while(expression)
{
   //الكود البرمجى المراد تنفيذة
}
```

ومن خلال الجملة السابقة فان الكود سيتم تنفيذة داخل الحلقة طالما أن الشرط expression صحيح

مثال للتوضيح :

```
public class Test {

   public static void main(String args[]) {

      int x = 0;

      while( x < 5 ) {

         System.out.print("ahmed");

         x++;
      }
   }
}
```

من خلال المثال السابق فان البرنام سيقوم بطاعة الاسم ahmed ثم يقوم بإضافة القيمة 1 الى المتغير x حيث أن الجملة ;x++ تعنى زيادة عدد واحد للمتغير ....ثم يعاود الكرة حتى يصبح قيمة المتغير x أكبر من 5 حينها سيصبح الشرط غير صحيح x < 5 و بالتالى ستتوقف الحلقة التكرارية عن التكرار ....و النتيجة أنه سيتم طباعة الاسم ahmed خمس مرات

- الحلقة التكرارية do...while

وتلك الحلقة تشبة كثيرا الحلقة while و الفرق بينهما أن الاولى تقوم بالتحقق من قيمة الشرط قبل تنفيذ ما بداخلها من تعليمات فى حين أن الثانية تقوم بتنفيذ التعليمات مرة واحدة أولا ثم تتحقق من الشرط و يتم بنائها على النحو التالى :

```
do
{
   //التعليمات البرمجية//

}while(Boolean_expression);
```

مثال للتوضيح :

```
public class Test {

   public static void main(String args[]){

      int x = 0;

      do{

         System.out.print("ahmed");

         x++;

      }while( x < 5 );
   }
}
```

وهنا سيتم ايضا طباعة الاسم أحمد خمس مرات .... وهى نفس النتيجة التى توصلها اليها فى الحلقة while و ذلك لكون أن الشرط صحيح 5 أكبر من قيمة المتغير x لكن فى حالة أن الشرط غير صحيح بمعنى أن 5 أقل من القيمة x فإن فى تلك الحالة تختلف النتيجة ما بين الحلقة الكرارية الاولى while و الحلقة التكرارية الثانية do...while فالاولى لن يتم تنفيذة أى كود برمجى بداخلها فى حين فى الثانية سيتم طباعة الاسم ahmed مرة واحدة ثم ينعدم الشرط فتقف عن الدوران

- الحلقة التكرارية for

و تستخدم تلك الحلقة التكرارية فى حالة معرفة عدد مرات الدوران أو التكرار المطلوب تنفيذها و يتم بنائها على النحو التالى :

```
for(initialization; expression; update)
{
   //التعليمات البرمجية المراد تنفيذها
}
```

وكما ترى أخى الكريم فإن جملة التكرار for تتطلب ثلاث معطيات على الاقل وهما :

Initialization .... المتغير الذى سنحقق به الشرط
Expression ..... الشرط
Update .... معدل الزيادة و النقصان

مثال توضيحى :

```
public class Test {

   public static void main(String args[]) {

      for(int x = 0; x < 5; x = x+1) {

         System.out.print("ahmed");

         System.out.print("\n");
      }
   }
}
```

- **الحلقة التكرارية enhanced For**

من إصدار الجافا رقم 5 دعمت نوع جديد من الحلقات التكرارية كوسيلة أبسط لتكرار عناصر تعبير(مجموعات البيانات) كالمصفوفات أو ArrayList ......الخ غير ذلك و يتم بنائها على النحو التالى :

```
for(مجموعة البيانات : متغير من نفس نوع مجموعة البيانات)
{
   //التعليمات البرمجية
}
```

مثال للتوضيح :

```
public class Test {

   public static void main(String args[]){

      int [] numbers = {10, 20, 30, 40, 50};

      for(int x : numbers ){

         System.out.print( x );

         System.out.print(",");
         System.out.print("\n");
      }

   }
}
```

مثال أخر :

```
public class Test {

   public static void main(String args[]){

      String [] names ={"James", "Larry", "Tom", "Lacy"};

      for( String name : names ) {

         System.out.print( name );

         System.out.print(",");
      }
   }
}
```

الفصل السادس : المصفوفات Arrays !!

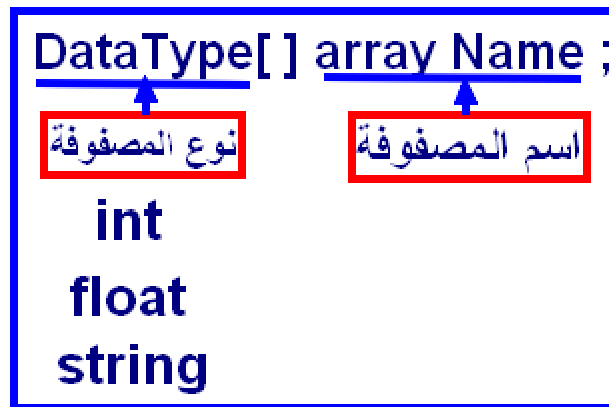
أولا : تعريف المصفوفة و انواعها:

المصفوفة هى عبارة عن مواقع يتم تخزين البيانات فيها لمدة مؤقتة طيلة فترة تنفيذ البرنامج فقط ..أو بمعنى عامى عبارة عن جدول يتم تخزين فية البيانات و لفترة معينة.......يوجد نوعان من المصفوفات:

- **المصفوفة الأحادية:** وهي مكونة من بعد واحد فقط.
- **المصفوفة متعددة الأبعاد:** وهي مكونة من عدد من الصفوف والأعمدة (ليس شرطا ان تكون بعدين.)

ثانيا : كيفية انشاء المصفوفات فى لغة الجافا

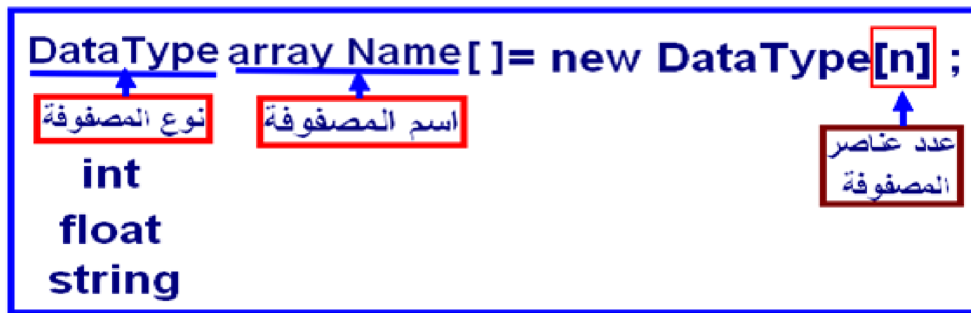
- **المصفوفة ذات البعد الواحد**....والصيغة العامة للإعلان عن المصفوفة ذات البعد الواحد كالتالي:



أو يمكن الإعلان عنها بنفس الصيغة السابقة مع وضع الأقواس بعد اسم المصفوفة كالتالي:



وبعد الإعلان عن المصفوفة لابد من تحديد عدد عناصرها ويتم ذلك كالتالي:



فمثلا لعمل مصفوفة رقمية من النوع int خاصة بدرجات عشرة طلاب مثلًا يتم ذلك كالتالي:

```
int array1[];
array1[] = new int[10];
```

أو يمكن الإعلان عن المصفوفة وتحديد عدد عناصرها في سطر واحد كالتالي:

```
int array1[] = new int[10];
```

أو بهذا الشكل التالى:

```
int[] array1 = new int[10];
```

أذن كل الخيارات متاحة امامك فى الاعلان و انشاء مصفوفة ذات بعد واحد

مثال للتوضيح : المطلوب عمل مصفوفة ذات بعد واحد تحتوي على درجات خمس طلاب وطباعة الناتج على الشاشة.بنفترض أن درجات الطلاب (١٠، ٢٠، ٣٠، ٤٠، ٥٠).

الحل و كتابة البرنامج كالتالى :

```
import java.util.Scanner;
public class array{

public static void main(String s[] ){
    int student[]={10,20,30,40,50};
        for(int i=0;i<=4;++i)
            System.out.println(student[i]) ;
        }
    }
```

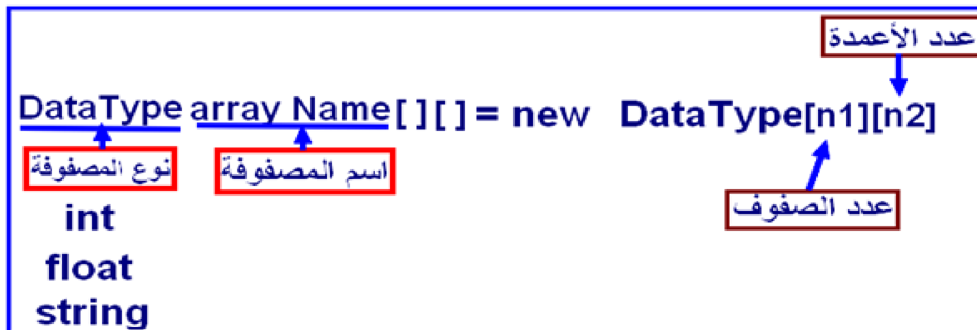
الأعلان عن المصفوفة واسناد قيم لها

حلقة تكرارية لقراءة محتويات المصفوفة

:

- المصفوفة متعددة الأبعاد (ذات البعدين)Multidimensional array

ويمكن القول بأن المصفوفة ذات البعدين هي عبارة عن جدول يحتوي على صفوف وأعمدة .والصيغة العامة لهذه المصفوفة كالتالي



ثالثا : كيفية أدخال العناصر الى المصفوفة من لوحة المفاتيح فى البيئة غير الرسومية

فى البيئة غير الرسومية :نقم بأستخدام دالة الادخال.... Scanner

فمثلا لنفرض أن هناك مصفوفة ذات بعدين يراد فيها إدخال درجات ٦ طلاب عن طريق لوحة المفاتيح يتم ذلك كالتالي:

سنقوم بتسمية المصفوفة student وسوف نستخدم دالة الإدخال Scanner لإدخال قيماً صحيحة إلى هذه المصفوفة وهذا هو شكل البرنامج

```
import java.util.Scanner;
public class array{

public static void main(String s[ ]){

1  int student[ ][ ]=new int[3][2];        ← الأعلان عن مصفوفة ذات بعدين

2  Scanner Keyboard=new Scanner(System.in);

3  for(int row=0;row<3;row++){             ← يتم استخدام حلقتين واحدة للصف
                                              والأخرى للعمود
4   for(int column=0;column<2;column++)

5  student[row][column]=Keyboard.nextInt() ;   ← يتم هنا ادخال عناصر المصفوفة

         }
      }
   }
```

:

في السطر الأول يتم الإعلان عن مصفوفة ذات بعدين من النوع int وعدد عناصرها ٦ عناصر.

السطر الثاني سبق وتم شرحه في جملة الإدخال.

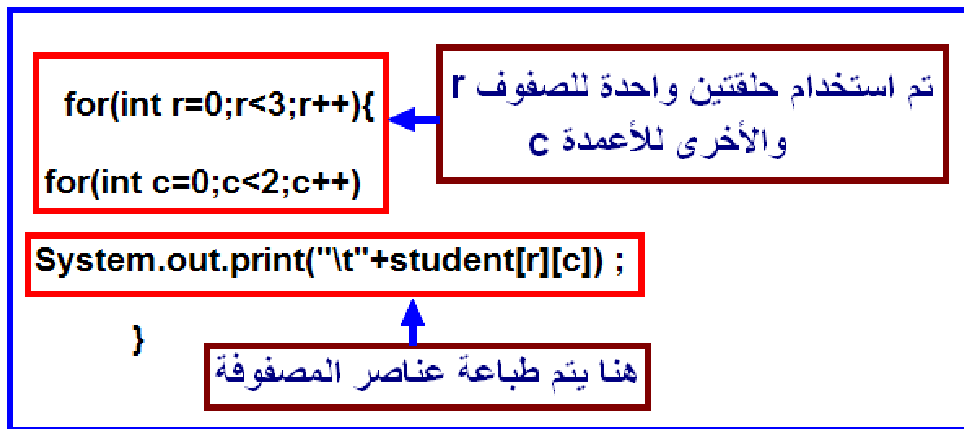
السطر الثالث والرابع تم عمل حلقتين ، الحلقة الأولى الخارجية للإشارة إلى رقم الصف row والحلقة الثانية للإشارة إلى رقم العمود .column وطبعا لاحظ ان الحلقة الخارجية قد تم إضافة أقواس لها .

أما السطر الخامس فيتم استخدام دالة الإدخال كما شرحناها سابقًا. ولكننا هنا استخدمنا المصفوفة student كمتغير يتم تخزين عناصر المصفوفة المدخلة فيه.

وعند تشغيل البرنامج لن تظهر لك أي رسالة . لأننا لم نضف رسالة للإدخال يمكنك أنت إضافتها ولكن، سيظهر المؤشر في أقصى يسار الشاشة منتظرا إدخال قيم عناصر المصفوفة فيتم إدخال ٦ عناصر والضغط على مفتاح الإدخال كل مرة. وبعد تمام الإدخال للستة عناصر تظهر علامة المحث للدرس دلالة على انتهاء الإدخال.

رابعا : كيفية قراءة عناصر المصفوفة

بعد أن يتم إدخال عناصر المصفوفة وإجراء أي عمليات عليها كالعمليات الحسابية مثلًا، يهمنا أن نرى المخرجات على الشاشة. وفي هذا البرنامج سوف نتعرف على كيفية قراءة عناصر المصفوفة وطباعة هذه العناصر كما هي على الشاشة دون أي تغيير، ويتم ذلك عن طريق البرنامج التالي:



خامسا : إدخال العناصر الى المصفوفة فى البيئة الرسومية

فى بيئة النوافذ أو البيئة الرسومية نستخدم JOptionPane و التى من خلالها نقوم بإنشاء مربع حوارى للمستخدم يستطيع من خلالة ادخال البيانات ....

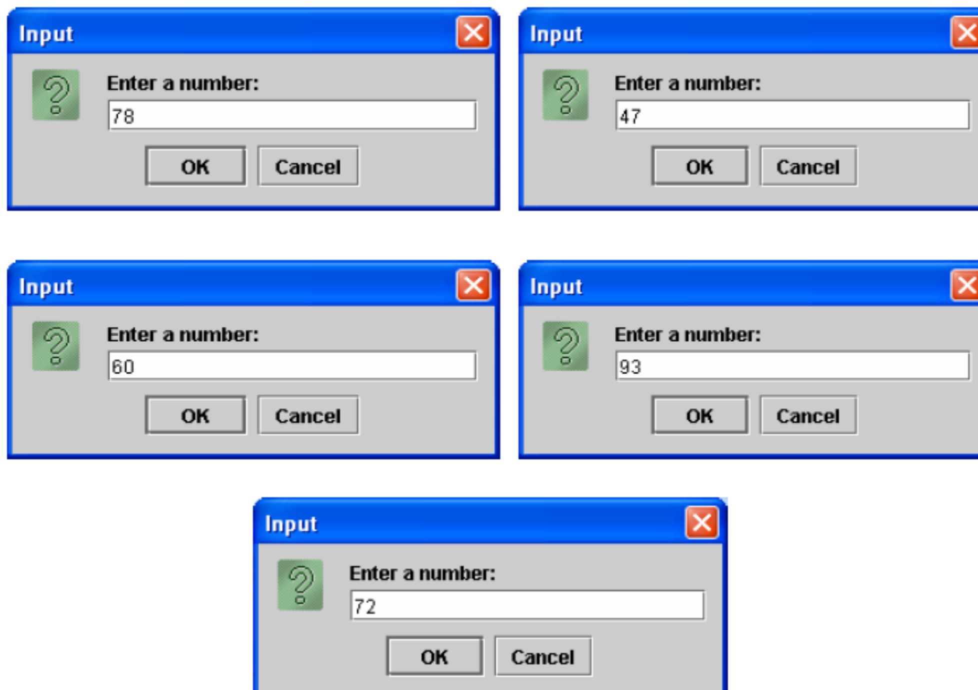
فلنتفرض مثلا أننا نريد انشاء مصفوفة تتكون من خمس عناصر يتم تخزين فيها مجموع الدرجات للطلاب ثم يقوم بعض النتائج التى أكبر من ٦٠ درجة .....الحل سيكون الكود على النحو التالى
:

```java
package array2;
import javax.swing.*;
public class Array2 {

    public static void main(String[] args) {
       int b[]=new int[5];
 String s;
 for(int i=0; i<5; i++){
 s=JOptionPane.showInputDialog("Enter a number:");
 b[i]=Integer.parseInt(s);
 } // end for
 for(int i=0; i<5; i++)
     if(b[i]>=60)
 System.out.println(b[i]);

    }
}
```

شرح المثال : فى السطر رقم (٤) قمنا بتعريف مصفوفة أسمها b وتم حجز ٥ مواقع فى تلك المصفوفة ...وفى الاسطر من (٦ الى ٩) قمنا بعمل حلقة تكرارية لإخراج المربع الحوارى input لادخال بيانات من المستخدم و ذلك لتخزينها فى مواقع المصوفة ...بعد ذلك يقوم البرنامج بطاعة الدرجات التى تساوى أو تزيد عن الرقم ٦٠ و يكون مخرجات البرنامج كما يلي :



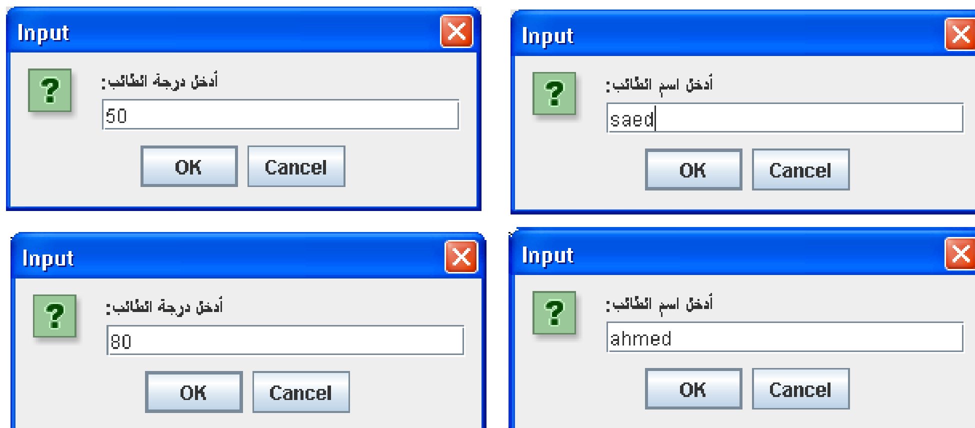
مثال أخر:

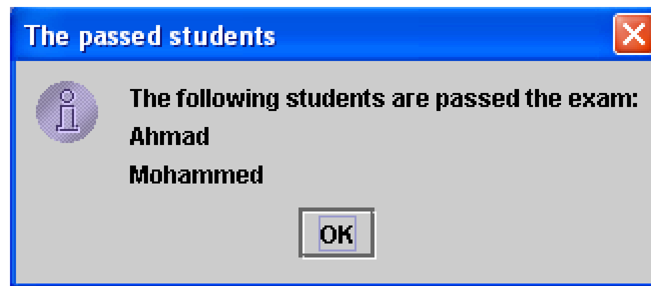
```java
package array2;
import javax.swing.*;
public class Array2 {

  public static void main(String[] args) {
  double marks[] = new double[6];
 String names[] = new String[6];
 String s;

 for(int i=0; i<6; i++){
 s=JOptionPane.showInputDialog("أدخل اسم الطالب:");
 names[i]=s;
 s=JOptionPane.showInputDialog("أدخل درجة الطالب:");
 marks[i]=Double.parseDouble(s);
 }
 String title = "The passed students";
 String results="The following students are passed the exam:\n";
 for(int i=0; i<6; i++){
 if(marks[i]>=60)
 results=results+names[i]+"\n";
 }
 JOptionPane.showMessageDialog(null, results, title,
JOptionPane.INFORMATION_MESSAGE);
 System.exit(0);
 }
}
```

الشرح : فى السطر رقم (٧) قمنا بإنشاء مصفوفة اسمها marks من نوع double بها ٦ مواقع وذلك لتخزين درجات الطلاب فيها ...أما فى السطر رقم (10) انشأنا مصفوفة أخرى اسمها names من نوع String لتخزين اسماء الطلاب ٦ بها ...من بداية السطر رقم (١١) استخدمنا الدوران لادخال اسماء و درجات الطلاب الستة و هذا طبعا عن طريق الامر showInputDialog الخاص بالكلاس.... JOptionPane ومن خلال الدروان ايضا يتم اضافة الطلاب التى تزيد أو تساوى درجاتهم الرقم (٦٠) الى المخرجات و طباعة المخرجات عن طريق الامر showMessageDialog الخاص ايضا بنفس الكلاس JOptionPane





خامسا : ترتيب عناصر المصفوفة :

فى كثيرا من التطبيقات نحتاج الى ترتيب المصفوفات أما ان يكون ترتيبا تصاعديا من الاكبر الى الاصغر أو ترتيبا تنازليا من الاصغر الى الاكبر ....وتوجد عدة طرق لذلك:

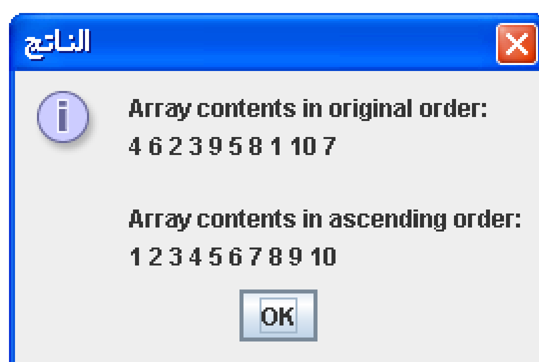
الترتيب المصفوفة بأستخدام خوارزمة الترتيب الفقاعى:

وتتم عملية الترتيب الفقاعي للمصفوفة بمقارنة كل عنصر مع العنصر التالي له، فإذا كان العنصر الأول أكبر من العنصر التالي تتم عملية تبديل العنصرين باستخدام متحول وسيط، وفي كل مرحلة نحصل على العنصر الأكبر في نهاية المجموعة المقارنة، وتكرر العملية هذه (n-1) مرة باعتبار أن (n) هو عدد عناصر المصفوفة.
والبرنامج التالي يبين ذلك :

```
1
2   package array2;
3   import javax.swing.*;
4
5   public class Array2 {
6
7     public static void main(String[] args) {
8
9    int num[] = new int[] {4, 6, 2, 3, 9, 5, 8, 1, 10, 7};
10   int temp;
11    String output="Array contents in original order:\n";
12   for(int i=0; i<num.length; i++)
13   output+=num[i]+" ";
14   for(int i=1; i<num.length; i++)
15   for(int j=0; j<num.length-1; j++)
16   if(num[j]>num[j+1]){
17   temp=num[j];
18   num[j]=num[j+1];
19   num[j+1]=temp;
20   }
21   output+="\n\nArray contents in ascending order:\n";
22   for(int i=0; i<num.length; i++)
23   output+=num[i]+" ";
24   JOptionPane.showMessageDialog(null, output, "الناتج",
25  JOptionPane.INFORMATION_MESSAGE);
26   System.exit(0);
27   }
28  }
```

فى الاسطر من ١٢ الى ٢٠ يوجد عمليتان دوران متداخلتان يتم من خلالها ترتيب عناصر المصفوفة num

حيث يتم فحص القيم فى السطر رقم ١٦ و اذا تحقق هذا الشرط يتم يتم تبديل عنصرين من عناصر المصفوفة بحيث بحيث يأخذ كل واحد من العنصرين مكان الاخر داخل المصفوفة ...وتكون مخرجات البرنامج كما يلى :



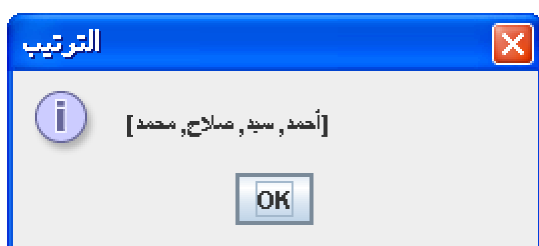
ترتيب المصفوفة بأستخدام الكلاس util.

الامر سهل جدا فى ترتيب عناصر مصفوفة ما بأستخدام الكلاس المسمى util كل ما عليك فعله هو استيراد هذا الكلاس الى مشروعك و البرنامج التالى يبين ذلك :

```
1
2   package array2;
3   import javax.swing.*;
4   import java.util.Arrays;
5   public class Array2 {
6
7     public static void main(String[] args) {
8
9    String names[] = {"صلاح" ,"سيد" ,"محمد","أحمد"};
10   Arrays.sort(names);
11   JOptionPane.showMessageDialog(null,Arrays.toString(names),"الترتيب",
12  JOptionPane.INFORMATION_MESSAGE);
13   System.exit(0);
14   }
15   }
```

الشرح:

فى السطر رقم (٤) قمنا باستيراد الكلاس... util.Arrays أما فى السطر رقم ٩ قمنا بإنشاء المصفوفة names و اعطاء القيم لها...أما رقم ١٠ قمنا بترتيب عناصر المصفوفة بأستخدام الامر Arrays.sort و وضعنا اسم المصفوفة المراد ترتيب عناصرها بين قوسين ...اما مخرجات البرنامج عن طريق المربع الحوارى showMessageDialogوهو يحيتوى على عناصر المصفوفة مترتبة كما فى الشكل التالى :



واخيرا تبقى لنا فى هذا الدرس كيفية البحث الخطى و البحث الثنائى فى المصفوفات فالى اللقاء

الفصل السابع : جمل معالجة الاستثناءات

من الثابت بقواعد الرياضيات انه لا يجوز القسمة على صفر بأي حال من الأحوال لذلك لم يتمكن البرنامج من تنفيذ العملية المطلوبة وستظهر للمستخدم رسالة خطا تنفيذي لا يمكنه تفاديها ولعل الأغلب أنها ستسبب بإغلاق البرنامج.

ولمعالجة أخطاء التنفيذ هذه وتحجيمها قبل حدوثها وفرت لنا لغات البرمجة وعلى رأسها الجافا آلية تتمثل بوضع جمل استثناءات خاصة يتوقع من خلالها المبرمج أنواع أخطاء التنفيذ التي قد تحدث ويقوم ببرمجة حدث برمجي بكل استثناء ليتم تنفيذه بدلا من خطا التنفيذي. الخالصة إن الاستثناءات هي عبارة عن آلية برمجية لمعالج أخطاء التنفيذ المحتمل وقعها**.....وهذا ما يقصد بمعالجة الاستثناءات فى لغة الجافا.**

أقسام جمل معالجة الاستثناءات

تقسم جملة معالجة الاستثناءات إلى ثلاثة أقسام رئيسية، هي:

الاول القسم try

وهي عبارة عن كتلة برمجية خاصة يكتب بداخله المبرمج التعليمات البرمجية التي يتوقع حدوث خطا تنفيذي خلالها وتكون صيغتها العامة على النحو الآتي:

الرمز:

```
try
{
  java code statements;
}
```

الثانى القسم catch

ويختص هذا القسم باحتواء التعليمات البرمجية التي من شانها معالجة الأخطاء التي قد تحدث داخل القسمtry وبما ان أخطاء التنفذ كثيرة متعددة فيمكننا صياغة أكثر من تعليمة catch لتحتوي كل واحدة منها على نوع معين من أخطاء التنفيذ وتكن عادة جميع جمل catch مرتبطة بجملة try تكتب جمل catch وفق الصيغة العامة التالية:

الرمز:

```
catch (ExceptionType name)
{
  java code statments;
}
```

يجب تضمين جملة catch نوع الخطأ المتوقع حدوثه Exception Type ومنح الخطأ اسما حتى نتمكن من التعامل معه. ويمكننا داخل كتلة catch صياغة تعليمات برمجية ليتم تنفيذها عند حدوث الخطأ لتضمن عند تنفيذ البرنامج تحجيم نسبة أخطاء التنفيذ وإعادة توجيه سير تدفق البرنامج إلى التعليمات المدرجة داخل catch دون ظهور رسالة خطا التنفيذ أو ان يؤدي ذلك إلى إغلاق البرنامج.

يعتبر القسم catch إجباري فلا يمكننا استخدام معالجة الاستثناءات بدونه ولكن يمكننا إدراجه بصورته العامة دون صياغة أي تعليمة برمجية بداخله.. وهنا بمجرد حدوث الخطأ التنفيذي المحدد داخل جملة catch الفارغة فان البرنامج سيتجاهل تنفيذ التعليمة البرمجية التي تحتوي الخطأ والمدرجة داخل القسم try ليتنقل مباشرة إلى تنفيذ التعليمات التي تلي معالجة الاستثناء كما سنبين ذلك في الأمثلة القادمة.

الثالث القسم finally علما بان الصيغة العامة للقسم finally تكون على النحو التالي.

الرمز:

```
finally
{
   java code statements;
}
```

بالرغم من إن القسم finally يعتبر اختبارا طالما أن القسم catch موجود إلا انه يعتبر أهم أقسام كتلة معالجة الاستثناء فمن خلاله يمكن للمبرمج تحديد أي من جمل الاستثناء catch التي سيتم تنفيذها إذا كان هناك أكثر من جملة. كما يمكن للمبرمج صياغة تعليمات برمجية داخل هذا القسم ليتم تنفيذها بصورة مستقلة عن التعليمات المدرجة داخل جمل catch كما يمكننا الاعتماد على القسم finally كبديل للقسم catch خاصة إذا كنا لا نريد تحديد نوع الخطأ المتوقع.

الصيغة الكاملة لكتلة معالجة الاستثناء

الشكل التالي يبين الأقسام الثلاث لكتلة معالجة الاستثناء

الرمز:

```
try
{
   java code statments;
}
   catch (ExceptionType1 name)
{
   Exception 1;
}
   catch (ExceptionType2 name)
{
   Exception 2;
}
finally
{
   java code statements;
}
```

آلية تنفيذ كتلة الاستثناءات

لمعرفة طريقة تنفيذ الكتل الاستثنائية، ادرس البرنامج التالي.

الرمز:

```
public class MyMain2
{

public static void main(Stringargs[])

{
```

```
system.Out.Println("Arabic SEO");
   int X=10;
   float b=X/0;
   System.Out.Println("The answer is"+b);

} }
```

عند ترجمة البرنامج السابق سنحصل على نتيجة ايجابية مفادها خلو البرنامج من الأخطاء القواعدية والتركيبية. ولكن النتيجة بعد تنفيذ البرنامج حتما ستكون مغايرة إذ سيظهر لنا معالج الأخطاء التنفيذية رسالة Run Time Error لاحظ ان البرنامج قام بتنفيذ تعليمة الطباعة Arabic SEO وبمجرد انتقال تدفق البرنامج إلى تعليمة القسمة float b= X/0 وجد معالج أخطاء التنفيذ أن عملية القسمة غير الشرعية فلا يجوز القسمة على صفر ولهذا السبب اظهر لنا رسالة خطا التنفيذ وبصورة متزامنة أوقف تنفيذ البرنامج ومنع تعليمة الطباعة الثانية من التنفيذ. لنعمل الآن على إعادة صياغة البرنامج باستخدام القسم tryمن معالجة الاستثناءات، كما في الشكل التالي:

الرمز:

```
public class MyMain2
{
   public static void main(String args[])
{
System.Out.Println("Arabic SEO");
   try
{
   int X=10;

   float b=X/0;
   System.Out.Println("The answer is"+b);

} }}
```

لاحظ عند ترجمة البرنامج سنحصل على خطا قواعدي ينص على إننا استخدمنا تعليمة الاستثناء try دون احد الأقسام الأساسية لها مثل catch أو finally , لذلك علينا الالتزام بالتركيب العام لجمل الاستثناء وإعادة صياغة البرنامج بتضمينها القسم catch لتصبح على النحو الآتي:

الرمز:

```
public class MyMain2
{
   public static void main (String args[])
{
System.Out.Println("Arabic SEO");
   try
{
   int X=10;

   float b=X/0;
   System.Out.Println("The answer is"+b);

}
```

```
catch (Exception e)
{

   System.Out.Println("Not Allow to Dived by 0");

}}}
```

استخدمنا القسم catch لمعالجة خطا القسمة على صفر، لاحظ إننا صرحنا عن نوع الخطأ التنفيذي بصورة عامة Exceptionدون أي تحديد. وقمنا بصياغة تعليمة طباعة لتظهر للمستخدم ولتوضح له إن القسمة على صفر غير مسموحة لها. عند تنفيذ البرنامج والوصول إلى تعليمة القسمة سيكتشف معالج أخطاء التنفيذ وجود الخطأ وبالتالي سينتقل مباشرة إلى القسم catch لينفذ ما بدخله من تعليمات. وبالتالي يمكننا إعادة صيغة البرنامج السابق بتوظيف القسم finally لطباعة النص Sorry أو أي كود برمجي آخر، كما في النص التالي.

الرمز:

```
public class MyMain2
{
   public static void main(String args[])
{
   System.Out.Println("Arabic SEO");
try
   {
   int X=10;
   float b=X/0;
   System.Out.Println("The answer is"+b);
}
   catch(Exception e)
{
   System.Out.Println(" ");
}
finally
{
   System.Out.Println("Sorry");
}}}
```

الفصل الثامن : التعامل مع الفئات Classes و الكائنات Objects و الطرق Methods !

في هذا الدرس نتناول حجر الزاوية في لغة الجافا آلا وهو الفئة clase حيث تعتبر الفئات الوحدات الأساسية في لغة البرمجة الكائنية نظرا لما تحتويه من كائنات وطرق وبيانات ويعتبر هذا الدرس من أهم الدروس في هذه الدورة حيث أن البرمجة الكائنية التوجه كان من المفروض أن تسمى البرمجة القائمة على الفئات ولكن ليس هذا وقت التعديل في مسميات أصبحت مشهورة في العالم بهذا المسمى الذي يبخس حق الفئات وينصف أحد مكوناتها وهي الكائنات ولكن ما علينا - هيا بنا نبدأ –

أولا ...تعريف الفئة: Classes

الفئة هي القالب البرمجي للغة الجافا الذي يحدد شكل الكائنات لما تحتويه من بيانات ومعاملات
والكائن ينتمي للفئة وليس هو كل الفئة وبالتالي للفئة عناصر منها المتغيرات والنهج (الدوال – الطرق)
إذا الفئة هي المكان الذي يستطيع المبرمجون أن يجمعوا فيه بين البيانات والطرق (الدوال) تحت سقف واحد

ثانيا ...إنشاء الفئات:

يتم إنشاء الفئة باستخدام الكلمة المفتاحية [ue]class وهي من الكلمات المحجوزة في الجافا ويكون الشكل العام للفئة كما يلي:

```
class Nameclass
}

{
```

هذا اذا كان برنامجنا يحتوى على عدة فئات مجتمعة فى ملف واحد و بالتالى فإن الفئة الرئسية التى تحتوى على الدالة main تبدأ بمحدد الوصول public class أى أن الفئة عامه ...وتبدأ جميع الفئات الاخرى بالكلمة المحجوزة class فقط كما أعلاه....
أما اذا كان برنامجنا يحتوى على عدة فئات كلا فى ملف منفصل فلابد من كل الفصائل تبدأ public class

ويجب أن تحتوى أى فئة على طريقة واحدة أو متغير واحد أو كلاهما وحين نريد أن نصمم فئة يجب أن نراعي نوع البيانات ومدى توافقها معا داخل الفئة بمعنى أن فئة مصممة لتخزين أسماء موظفين وأرقام هواتفهم وعناوينهم لن يكون من المناسب أن أضع في نفس الفئة بيانات عن حالة الطقس وحالة البحر ومعدل سقوط المطر ولذلك يجب أن نراعي الناحية المنطقية عند تصميم الفئة من حيث علاقة البيانات التي نريد التعامل معها في نفس الفئة

ثالثا ... الكائنات Objects

هى عناصر تمثيل استخدام الفئة فى البرنامج وهى تأخذ نفس بيانات الفئة من دوال وطرق و متغيرات ...الخ و الطريقة الوحيدة لانشاء الكائن تكون بإستخدام الامر.... new باستخدام الشكل العام التالى

```
classname objectname = new classname ();
```

وهذا يعنى أسم الفئة أسم الكائن = الكلمة المحجوزة (new) أسم الكلاس أو الفئة;()

مثال توضيحى / قم إنشاء برنامج لحساب العديدن ٥ و ٨ بأستخدام عدد ٢ كلاس او فئة ؟

الحل:

الرمز:

```
package ex1;

class Egy{
   int no1=5,no2=8,total;
     }

public class Ex1 {

 public static void main(String[] args) {

     Egy cal= new  Egy();

     cal.total=cal.no1+cal.no2;
     System.out.println("Value= "+cal.total);
   }
}
```

- **فى السطر الثانى...** قمنا بأنشاء الفئة أو الكلاس Egy عن طريق الامر class Egy ثم فتحنا قوسى البداية و النهاية ...ويجب ملاحظة أننا استخدمنا الكلمة المحجوزة class لانشاء تلك الفئة بدون محدد الوصول... public وذلك لكون جميع فئات برنامجنا فى ملف واحد فقط.
- **فى السطر الثالث....** نلاحظ أننا عرفنا المتغيرات الثلاث وهي [no1] لتحزين الرقم الاول و [no2] لتخزين الرقم الثانى و [total]لتخزين عملية الجمع....

و هكذا نلاحظ اننا قمنا بإنشاء ابسط كلاس او فئة لا تحتوى إلا على الاعلان عن المتغيرات فقط بدون اى طرق او دوال.

- **فى السطر الرابع...** أنشأنا الكلاس او الفئة الرئسية Ex1 و التى تحتوى على الدالة الرئسية main و يجب ملاحظة أن تلك الفئة أو الكلاس بدأت بمحدد الوصول public
- **فى السطر الخامس...** قمنا بإنشاء الكائن cal من الفئة Egy بإستخدام الكلمة المحجوزة new
- **فى السطور الاخرى....** بعد ان انشأنا الكائن cal نتسطيع بسهولة استخدام المتغيرات الموجودة فية و هى طبعا امتداد للبيانات و المتغيران من الفئة المنشىء منه هذا الكائن

رابعا : الطرق (الدوال) Methods

كما نو هنا سابقا عند الحديث عن الفئات بأن المكونات الرئيسية للفئة هي المتغيرات الأولية والطرق أو الدوال فإننا نتناول بالتفصيل في هذه الجزئية الحديث عن الطرق....

1_تعريف الطريقة [السلوك]

هي ذالك الجزء من الفئة الذي يختص بالتعامل مع البيانات وتعرف داخل الفئة و بها يتم التفاعل بين أجزاء البرنامج المختلفة والفئة و تحتوي الطريقة على جملة برمجية أو أكثر والكود البرمجي الجيد في لغة الجافا هو الذي يكتب بحيث تؤدي الطريقة مهمة واحدة ولكل طريقة اسم يتم استدعاء الطريقة به ويمكنك أن تطلق على طريقتك الاسم الذي تفضله مع التنبيه على عدم استخدام الكلمات المحجوزة في لغة الجافا لهذه التسمية...

٢_الشكل العام للطريقة

Access specifier **return type** method name(**parameter type** **parameter name**)

وقبل أن نناقش الشكل العام للطريقة بالتفصيل نذكر نوعا الطرق و هما

النوع الأول : هو النوع الذي يعود بقيمة ويستوجب استخدام return
النوع الثاني : هو النوع الذي لا يعود بقيمة void

والآن مع مناقشة الشكل العام المختصر للطريقة:

- **المقطع الاول Access specifier** محدد الوصول للطريقة أى نوع الوصول للطريقة فمثلا النوع العام الذي يتيح الوصول للطريقة من خارج الفئة ومن خارج البرنامج ويأخذ الكلمة المفتاحية public أما النوع الآخر وهو عكس النوع العام ويأخذ الكلمة المفتاحية private وهو يسمح بالوصول للطريقة من داخل الفئة فقط أما النوع الثالث فهو النوع المحمي ويأخذ الكلمة المفتاحية protected
- **المقطع الثانى return type** يحدد نوع القيمة المرتجعة من الدالة .
- **المقطع الثالث method name** وهو الذي يعطي للطريقة اسمها ومن الافضل استخدام أسم يدل على وظيفية الطريقة ودائما فى لغة الجافا يبدأ الاسم بحرف صغير ...وفى حالة الاسم الذى يتكون من اكثر من مقطع يبدأ المقطع الاول بحرف صغير و المقطع الثانى بحرف كبير... علما بأن هناك طرقا تعرف بواسطة المبرمج وأخرى تكون جاهزة توفرها لغة الجافا وما على المبرمج سوى استدعائها مثل الطرق الرياضية وحين تعرف طريقة يجب عليك إتباع الشروط الخاصة بالتعريف في لغة الجافا
- **المقطع الرابع- parameter** يوضع بين القوسين بيان بالعوامل الداخلة للطريقة ويفصل بين العامل والآخر فاصلة وفي حالة عدم وجود عوامل لا يوضع شيء بين القوسين
- المقطع الخامس جسم الطريقة.{ اى الاكواد المراد تنفيذها من خلال تلك الطريقة.

مثلا توضيحى : قم إنشاء برنامج لحساب العديدن ٥ و ٨ بأستخدام عدد ٢ كلاس او فئة مع إنشاء طريقة لحساب الناتج... الحل:

الرمز:

```
package ex1;

class Egy{
  int no1=5,no2=8,sun;

 void total(){
   sun = no1+no2;
   System.out.println("Value= "+sun);
 }
}
```

```
public class Ex1 {

  public static void main(String[] args) {
   Egy cal=new  Egy();

    cal.total();

    }
}
```

دعنا الآن نلقي نظرة سريعة على هذا البرنامج ولنبدأ ببيت القصيد ألا وهو إضافة الطريقة ومع السطر الأول

```
void total ( ) {
```

في هذا السطر نعلن عن طريقة بمسمىtotal بدون عوامل ولا ترجع قيمة void ولذلك لا ترجع قيمة لمن استدعاها ونهاية السطر توجد حاصرة للإذن ببدء جسم الطريقة الذي يتكون من سطرين هما

```
 sun = no1+no2;
   System.out.println("Value= "+sun)
```

ويتم تشغيل الطريقة فى الدالة الرئسية للبرنامج بعد أنشاء كائن من نفس الفئة عن طريق الامر التالى:

```
 cal.total();
```

مثال توضيحى أخر.. قم بحل نفس السؤال السابق ذكرة مع استخدام العوامل فى الطريقة total

```
class Egy{
  void total(int no1,int no2){
    int sun = no1+no2;
    System.out.println("Value= "+sun);
  }
}

public class Ex1 {
public static void main(String[] args) {

    Egy cal=new  Egy();

    cal.total(8,5);

    }
}
```

3_العائد من الطريقة Returning from aMethod

كما ذكرنا سابقا يوجد نوعان من الطرق هما:

النوع الأول : هو النوع الذي لا يعود بقيمة void method مثل الحالة في المثال السابق أعلاه

النوع الثاني : هو النوع الذي يعود بقيمة الطرق التي تعود بقيمة و هي الأكثر استخداما في مختلف مجالات البرمجة ويستوجب استخدام return و الشكل العام لها يكون على النحو التالي:

الرمز:

```
public int Meth03( ){

//Body of Method03
// طريقة لا تحتوي على عوامل وتعود بقيم
return 0;
}
```

مثال توضيحي: قم بحل نفس السؤال السابق ذكرة مع جعل الطريقة total ترجع بقيمة العملية الحسابية

```
class Egy{
   int no1=5,no2=8;

  int total(){

   return no1+no2;
  }
}

public class Ex1 {

  public static void main(String[] args) {

      Egy cal=new  Egy();

    System.out.println("Value= "+cal.total());

    }
}
```

نلاحظ فيه أننا قمنا بأنشاء الطريقة (total) الذي سوف يستقبل القيمة الراجعة التي سوف تكون هنا من النوع [int] وذلك يستدعي أن تعود بقيمة من النوع الصحيح لمن استدعاها وعلى سبيل المثال إذا أردت طريقة تعود بقيمة من النوع [double] يجب عليك اختيار هذا النوع ....كما يلاحظ أننا استخدمنا الامر (return) لارجاع قيمة المعادلة (no1+no2)...

4_المنشآت - البانيات Constructors

في المثال السابق تم كتابة المتغيرات بطريقة يدوية ....وكانت هذه الطريقة للتدريب ولكن لمحترفي البرمجة يوجد شيء آخر لانجاز هذا الأمر آلا وهو استخدام البانيات Constructors ولكي نوضح هذا المفهوم إليك هذا المثال

```
class Egy{
   int no1,no2;
  Egy(){
  no1=0;
  no2=0;
}
 int total(){

   return no1+no2;
  }
}

public class Ex1 {

    public static void main(String[] args) {

     Egy cal=new  Egy();

  cal.no1=5;
  cal.no2=8;
   System.out.println("Value= "+cal.total());

   }
```

مما سبق نستنتج أن البانيات تشبه الطريقة من حيث القواعد وتأخذ نفس اسم الفئة وفيها الحل لمشكلة تحديد القيم الابتدائية وحيث أن اسم الباني هو نفسه اسم الفئة فإن الباني سوف يستدعى تلقائيا........

ففي المثال أعلاه تم إنشاء فئة بسيطةclass Egy وكان الباني فيها هو Egy وكما نلاحظ فإن اسم الفئة يشبه تماما اسم الباني وعندما نقوم بإنشاء كائن cal حسب الجملة البرمجية (التعليمة)

```
Egy cal=new  Egy();
```

يتم حجز مكان لهذا الكائن ثم يطلب الباني وسيتم تحديد قيمة ابتدائية له ونلاحظ هنا أن الباني قد خصص القيمة (٠) لكل من المتغيرno1,no2 وأن هذا الباني قد تم استدعاؤه بواسطة new عند إنشاء الكائن Egy

وبالتالي نستنتج فوائد المنشآت (البانيات) التي تعتبر (طريقة) غي عادية حيث لا تمتلك قيمة راجعة وتخلصنا من المشاكل البرمجية الخاصة بالفئات التي تنجم عن الكتابة بشكل يدوي كما في المثال قبل التعديل وأيضا تسهل قراءة وفهم مصدر البرنامج البرمجي .....و الخلاصة أن:

البنا - نهج (طريقة) خاص يتم تنفيذه عندما يبدأ كائن الفئة ويحتوي من الوسطاء ما يحدد شكل الكائن ويتم استدعاؤه تلقائيا لأنه يحمل نفس اسم الفئة والبانيات لا تمتلك قيمة راجعة

الباب الثانى :واجهة المستخدم الرسومية

الفصل الاول : الاطارات أو النوافذ

درس اليوم أن شاء الله تعالى سيكون عن واجهة المستخدم الرسومية والتى تتمثل فى النوافذ و الازرار و مربعات النصوص و القوائم وغيرها..... وسنخصص هذا الدرس عن أهم العناصر الرسومية التى تستخدم فى برمجة التطبيقات بلغة الجافا و هى **الاطارات أو النوافذ**

أولا أنواع الاطارات و النوافذ فى الجافا

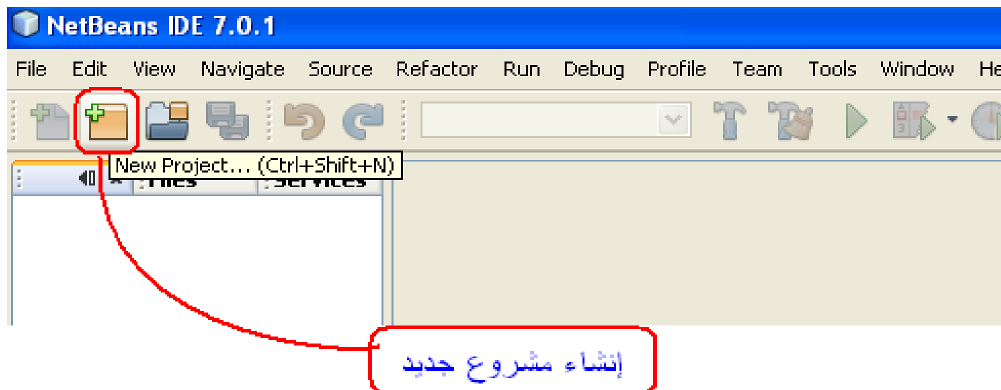
يوجد نوعين من الاطارات أو النوافذ فى لغة الجافا ....

- النوع الاول farme : وهو ينتمى للحزمة awt
- النوع الثانى jfarme وهو ينتمى للحزمة swing

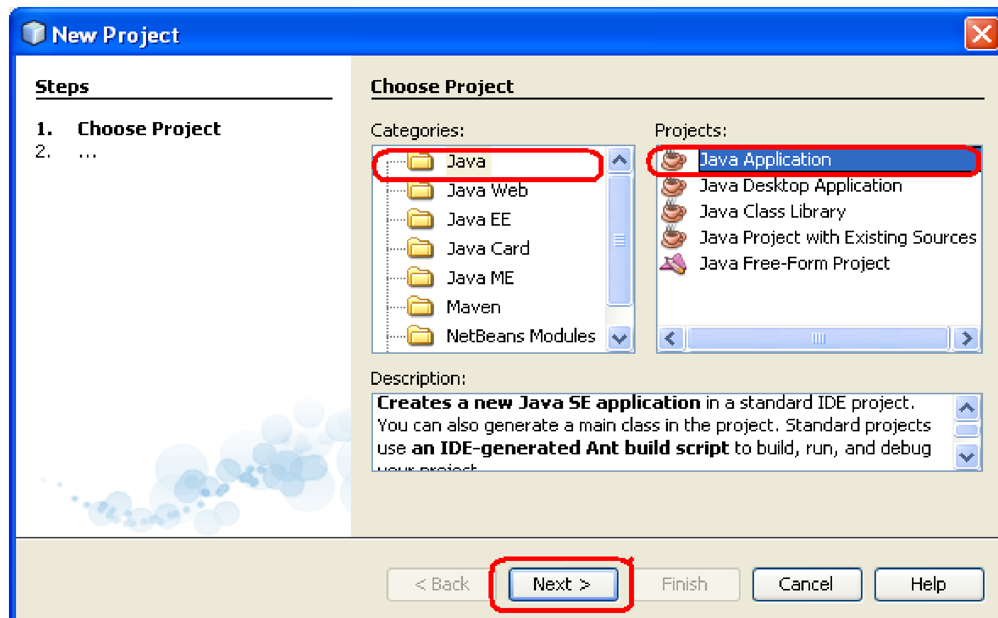
والفرق بينهما أن farme يأخذ شكل نظام التشغيل الذى يعمل عليه فى حين jfarme لا يتغير شكله بتغير نظام التشغيل

ثانيا _ كيفية إنشاء النوافذ التابعة للحزمة swing

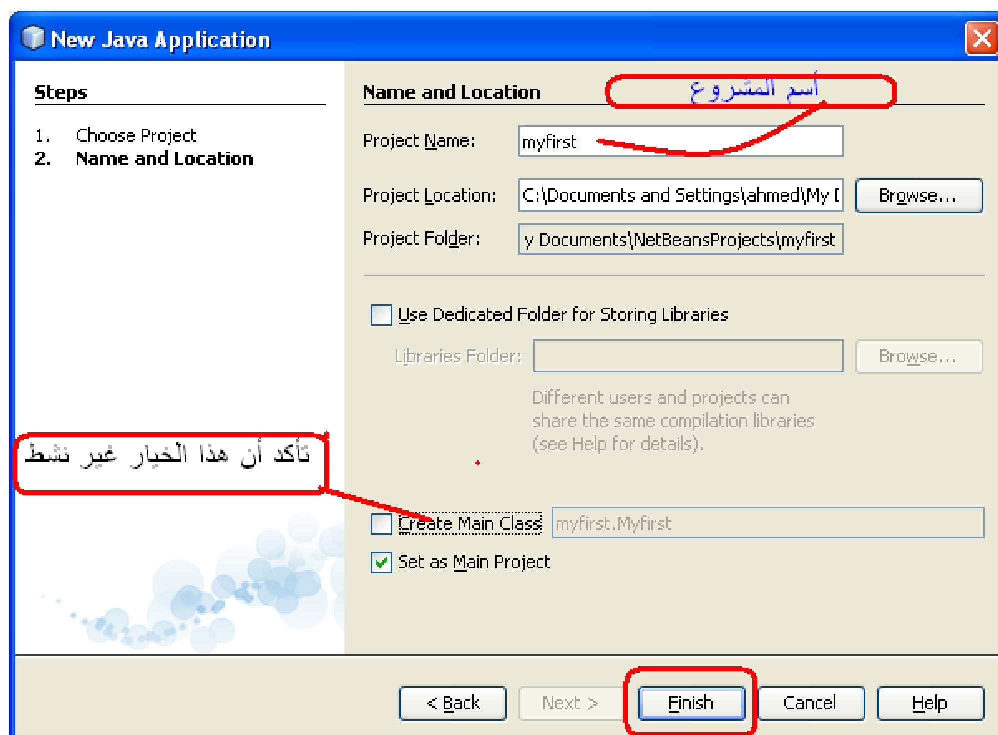
1_قم بأنشاء مشروع جديد و ذلك من خلال الضغط على ايقونة **مشروع جديد** الموجودة فى شريط الادوات كما فى الصورة التالية:



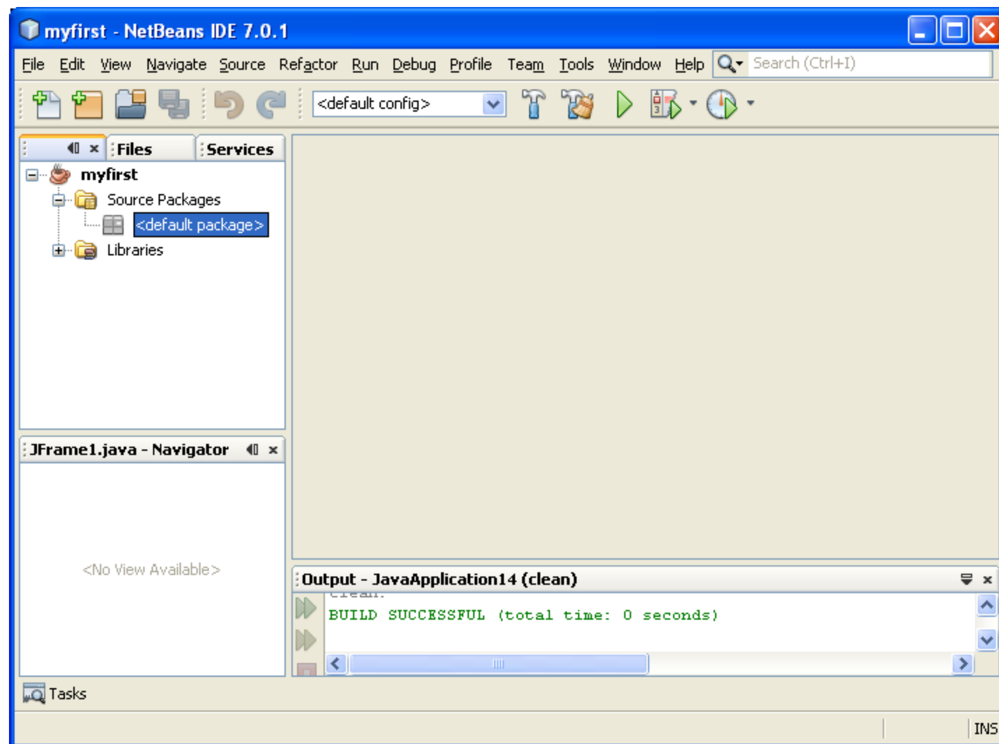
2_سوف تظهر لك نافذة **مشروع جديد** فى جانب التصنيف أختر **java** ستجد أن القائمة مشاريع بها عدة مشاريع حدد نوع المشروع **java application** كما فى الصورة التالية ثم نضغط **Next**



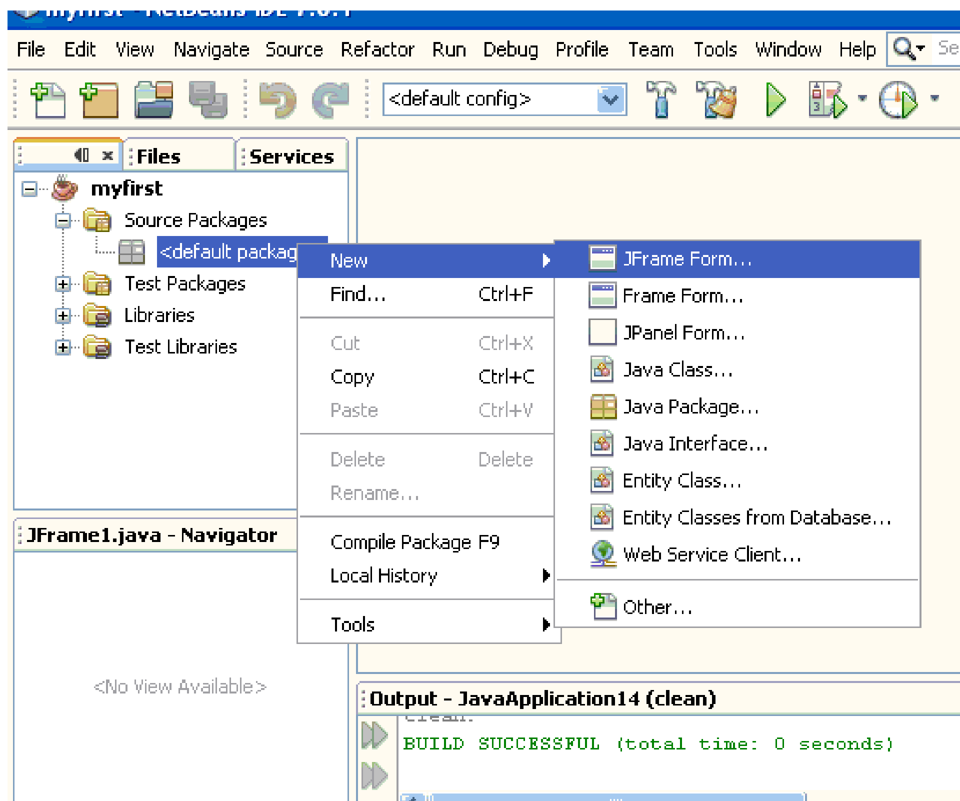
3_فى المربع الحوارى **New java application** قم بوضع اسم للمشروع وحدد المسار الذى تريد أن تحفظ فية مشروعك ...ثم تأكد أن الخيار **create main classe** غير نشط ثم اضغط **finsh**



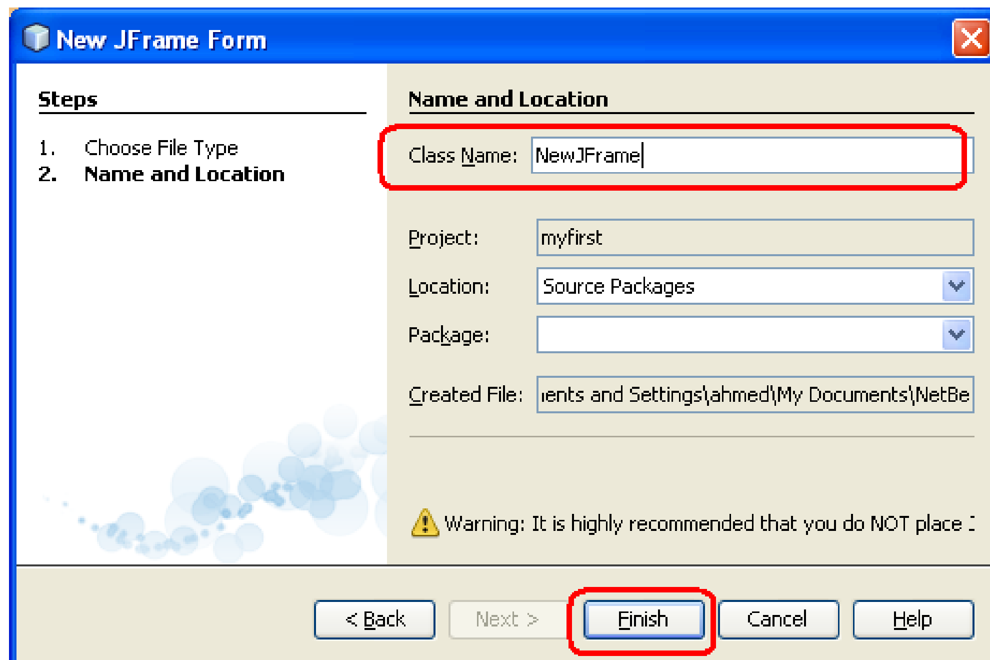
ستجد ان الصورة التالية:



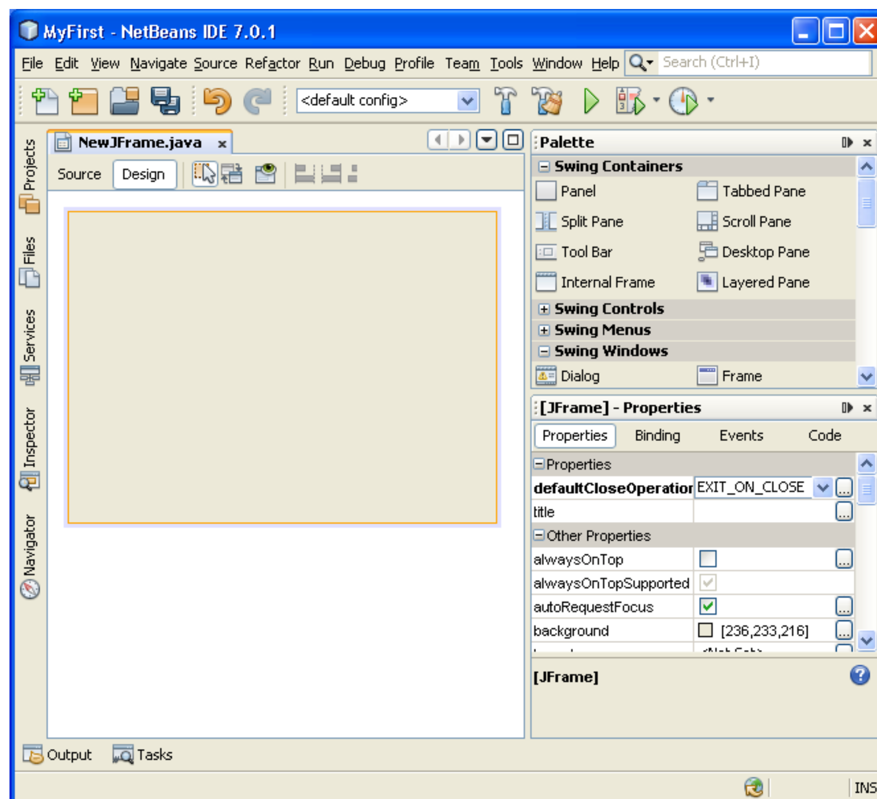
4_قم بالضغط دبل كليك على ايقون **source packages** ثم قف بالماوس على
ايقون default packages ثم كليك شمال بالماوس ثم New ثم jfarme كما فى الشكل التالى:



سوف يظهر لك المربع الحوارى التالى قم بإعطاء اسم للنافذة ثم اضغط إنهاء كما يلى:

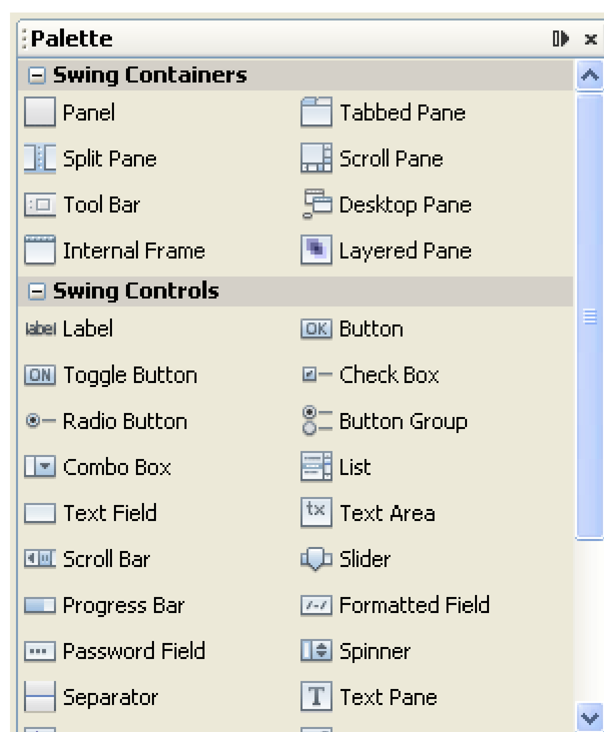


5_بعد الضغط على إنهاء كما فى الشكل السابق نكون قد قمنا بإنشاء نافذة و ستجدها كما فى الشكل التالى:

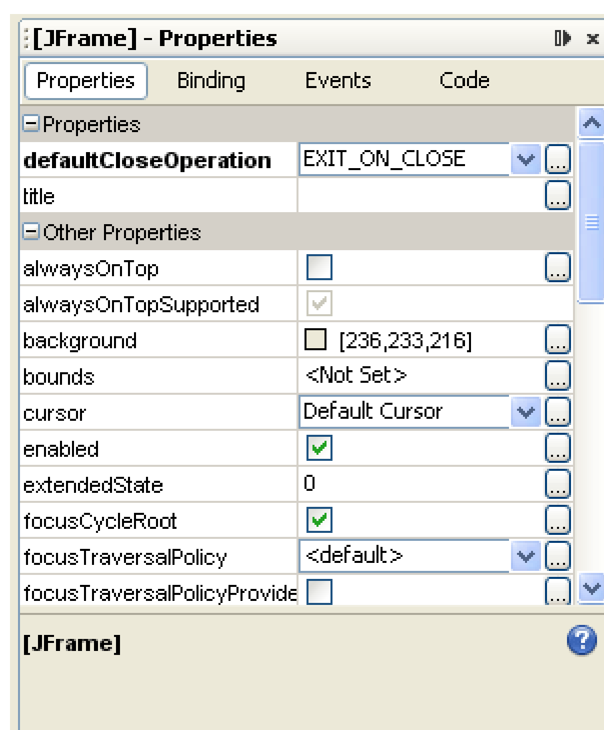


كما هو ظاهر فى الصورة أعلاه قد قمنا بأنشاء نموذج أو نافذة عن طريق الجافا ولا يحتوى على اى كائن اخر فقط مجرد نافذة تحتوى على ازار الاغلاق و التكبير و الاخفاء و لا تظهر إلا عندما يتم تشغيل النافذة .....و

ويوجد على يمين النافذة أو الفورم الذى قمنا بإنشاءه نافذتين ....الاولى تسمى **Palette** و هى تحتوى على جميع العناصر الرسومية الخاصة بالحزمةswing و الحزمة awt مثل الازرار و مربعات النصوص و القوائم وغيرها و سوف يتم شرح تلك العناصر فى دروس لاحقة............



أما النافذة الثانية هى نافذة **الخصائص** الخاصة بالفورم ومن خلالها نستطيع تغير بعض خصائص الفورم و هى كما فى الشكل التالى.



ثالثا : التعرف على أهم خصائص jfarme

بعد ما قمنا بإنشاء اول نافذة لنا أخوانى الكرام تعالوا الان نتعرف على اهم خصائص هذا الكائن.. jfarme

1_الخاصية : **default close operation** وهى خاصة بعملية أغلاق النافذة من خلال زر الاخلاق وتأخذ أربع قيم :

- HIDE و تؤدى الى أخفاء الفورم أو النافذة عن الضغط على زر الاغلاق معنى ذلك انها تكون محملة فى ذلكرة الجهاز و لكنها مخفية فقط
- EXIT_ON_CLOSE وتؤدى الى أغلاق التطبيق ككل بما فى ذلك باقى نوافذ التطبيق فى حالة الضغط على زر الاغلاق
- DO_NOTHING ولا تؤدى الى فعل اى شىء فى حالة قيام المستخدم الضغط على زر الاغلاق.
- DISPOSE تؤدى تلك الخاصية الى أخفاء و اغلاق النافذة الحالية فقط دون باقى نوافذ التطبيق.

2_الخاصية. title وهى الخاصية السئولة عن اعطاء عنوان للفافذة و هى تساوى caption فى الفجوال بيسك .

3_الخاصية... alweysOnTop تأخذ تلك الخاصية قمية true أو.... false ووضيفتها جعل النافذة أو الفورم ظاهرا فوق جميع النوافذ فى حالة تحديدها, true و العكس صحيح.

4_الخاصية..... resizable و ظيفة تلك الخاصية انه عند تحديدها يستطيع المستخدم تغير حجم النافذة أثناء التشغيل و العكس صحيح.

5_الخاصية.... name و تختص بإعطاء أسم برمجى للنافذة و الذى سوف تتعامل معه فى كتابة الاكواد

6_الخاصية..... type و تلك الخاصية تأخذ ٤ قيم منها Normal و تجعل النافذة بالشكل العادى مكون من زر الاغلاق و التكبير و التصغير UTILITY ,... وهى تجعل النافذة لا تحتوى الاعلى زر الاغلاق فقط,....

7_الخاصية..... undecorated و تلك الخاصية تقوم عمل نافذة بدون حواف

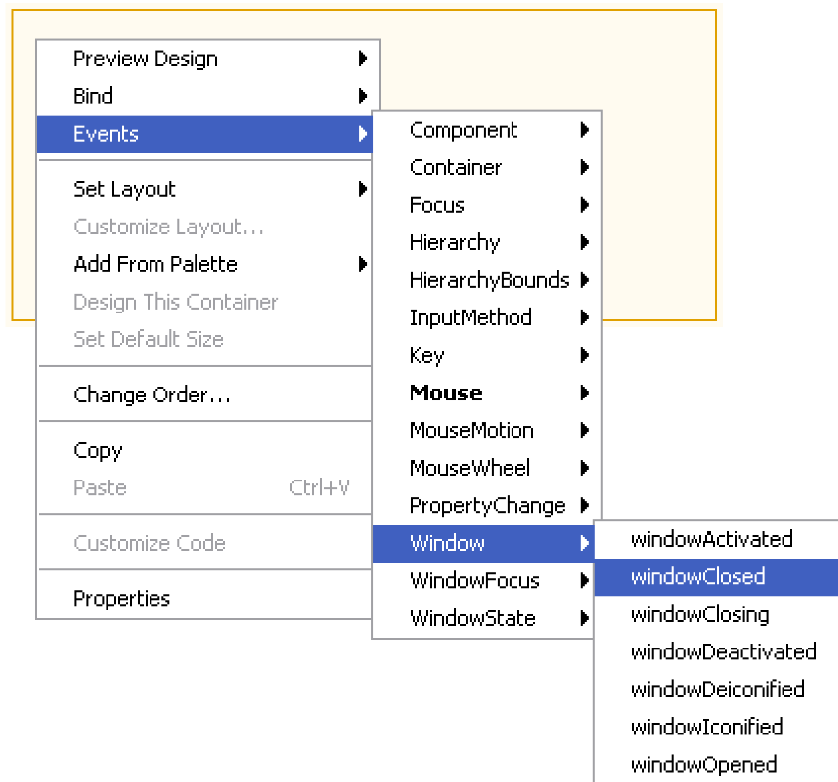
8_لتوسيط الفورم أو النافذة على الشاشة عليك النقر على التوبيب (Code) الموجود فى صفحة الخصائص ... porpertiseستجد form size policy قم بجعلها تساوى القيمة.... Genarate Resize code

كما يوجد العديد من الخصائص الاخرى التى تستطيع ان تكتشفها بنفسك

رابعا : أهم الاحداث الخاصة بانوافذ و الاطارات.

تعلمنا فى لغات برمجة اخرى مثل الفجوال بيسك أن الحدث هو الفعل المراد تنفيذ الكود عند تحقيقة ....فمثلا عند أغلاق النافذة نريد اظهار رسالة للمستخدم و بالتالى فأن أغلاق النافذة حدث وهكذا .....ويوجد عدة أحدث للنوافذ فى لغة الجافا ومن اجل اختيار حدث لكائن ما و ليكن jfarme الحالى مثلا وكتابه الكود بداخله نتبع ما يلى:

١ -نحدد الكائن المراد التعديل او العمل عليه.
٢ -ننقر بزر الماوس الايمن عليه.
٣ -سوف تظهر لنا قائمة كبيره نختار events
٤ -سوف تظهر لنا قائمه اخرى كبيره نختار منها تصنيف الحدث وليكن Window
٥ -سوف تظهر لنا قائمه اخرى كبيره نختار منها الحدث المطلوب وليك WindowClosing كما فى الشكل التالى :

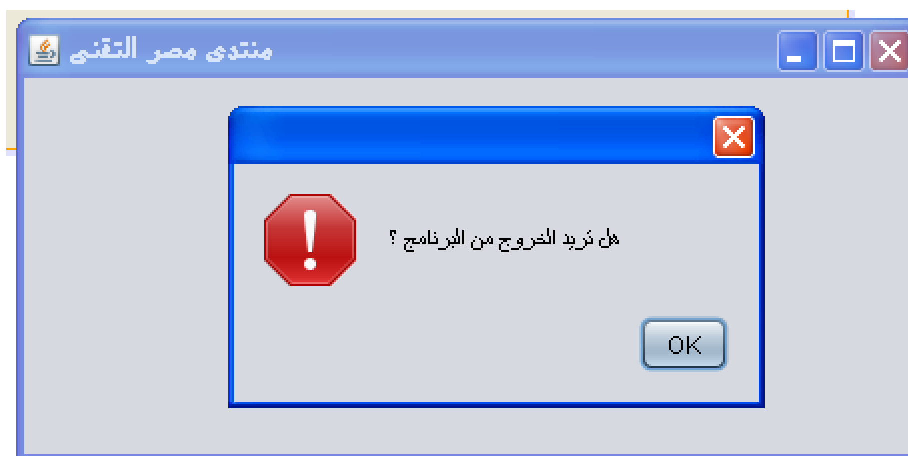


سوف يتم الانتقال بك الى صفحة الكود تحت الحدث المنشيء formWindowClosing لتقوم بكتابة الكود المراد تنفيذة عند هذا الحدث فمثلا نريد اظهار رسالة للمستخدم عند خروجه من البرنامج تفيدة بانه متأكد من الخروج من البرنامج و أم لا . .... سنقوم بكتابة الكود تحت الحدث formWindowClosing كما فى الشكل التالى :

تكبير الصورة معاينة الأبعاد الأصلية.

```
10      @SuppressWarnings("unchecked")
11 [+]  Generated Code
37
38 [-] private void formWindowClosing(java.awt.event.WindowEvent evt) {
39     JOptionPane.showMessageDialog(null,"O,"","هل تريد الخروج من البرنامج ؟");
40     }
41
```

والنتيجة انه عندما يتم اغلاق الفورم أو النافذة يتم تنفيذ الكود كما فى الشكل التالى :

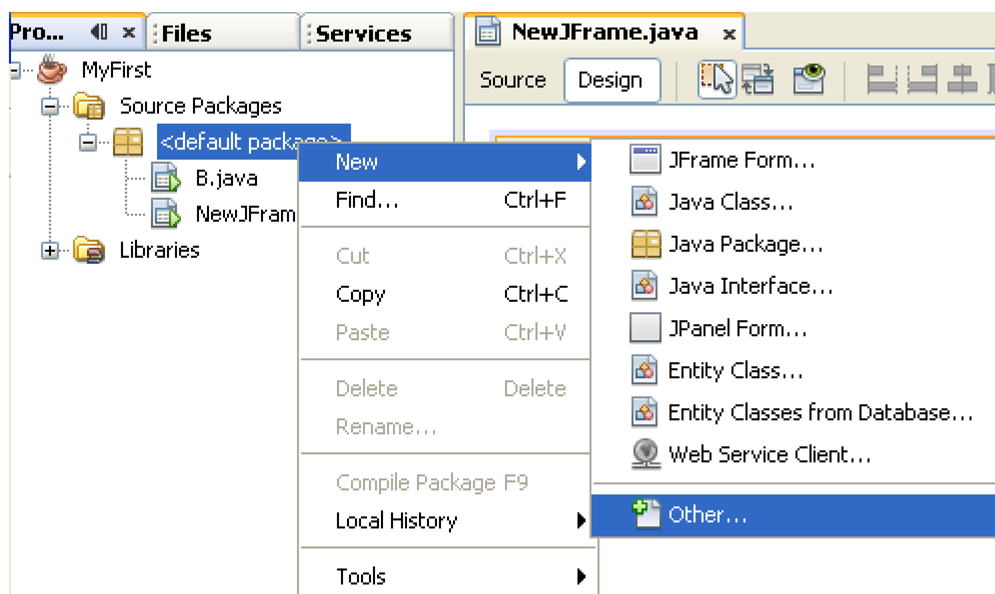


والان تعالوا نتعرف على أهم أحداث النافذة:

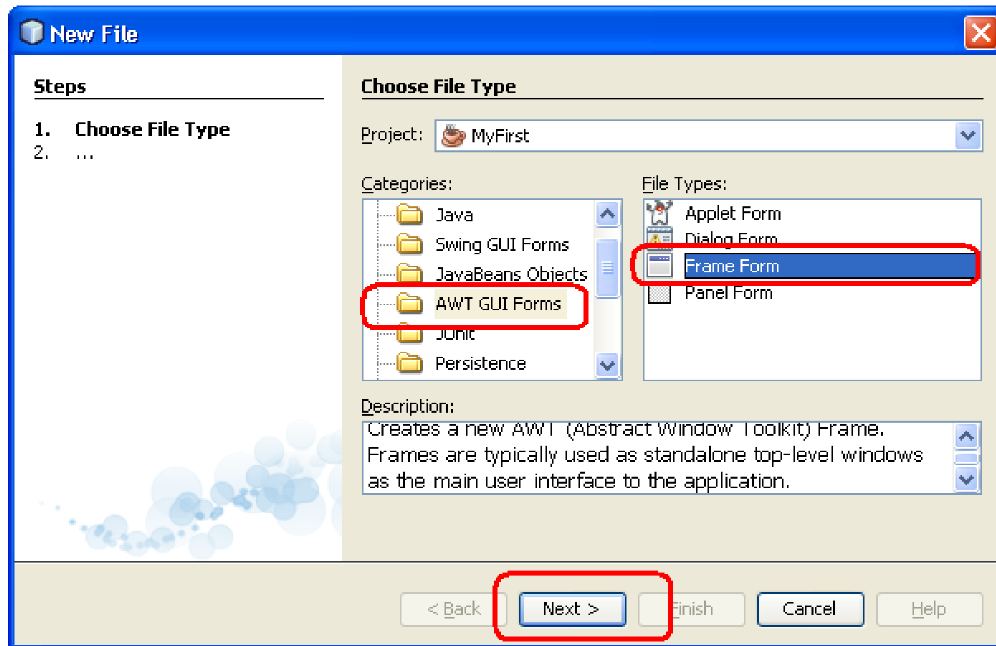
الحدث.....windowOpened و هو حدث فتح النافذة
الحدث.....windowClosing و هو حدث يستخدم اثناء بدء عملية أغلاق النافذة ...مثل اظهار رسالة قبل عملية أغلاق الفورم أو النافذة
الحدث....windowClosed و هو حدث يستخدم بعد عملية أغلاق الفورم وليس اثناء الاغلاق.
الحدث....windowActivated و هو حدث يستخدم فى حالة كون النافذة نشطة.
الحدث windowGainedFocus و الحدث....windowLostFocus و هو حدث يستخدم فى حالة كون النافذة التركيز أو فقد التركيز عليها
كما يوجد العديد من الاحداث خاصة بحركة الماوس على النافذة منها MouseClicked و ايضا احداث خاصة بمفاتيح الكيبورد و تستطيع ان تكتشفها بنفسك.

خامسا: أنشاء النوافذ الكلاسكية farme الخاصة بالحزمةawt

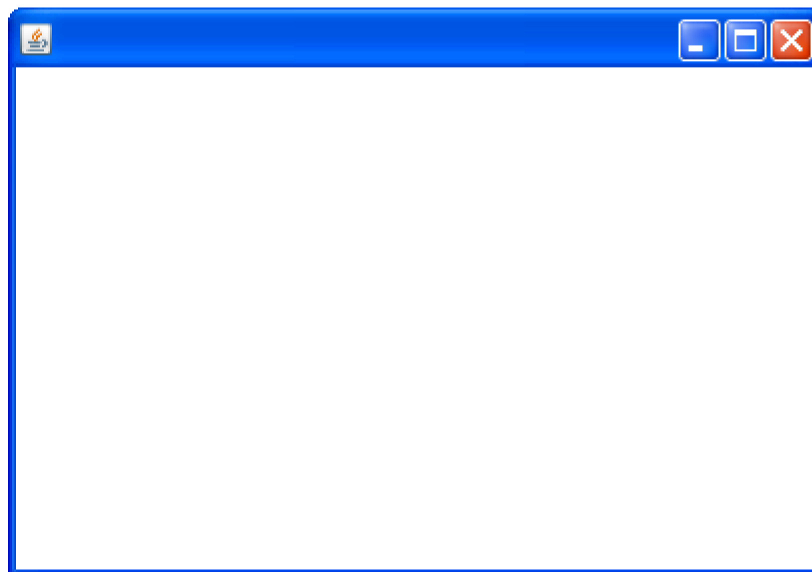
تحدثنا فى بداية الدرس انه يوجد نوعين من النوافذ فى لغة الجافا ....نوافذ تتبع الحزمة swnig وقد شرحنا كيفية انشائها و تعرفنا على اهم خصائصها ....الان دعونا نتعلم كيفية إنشاء النوافذ التى تتبع الحزمة. awt
_قف بالماوس على ايقون default packages ثم كليك شمال بالماوس ثم New ثم Other كما فى الشكل التالى:



_سوف يظهر لنا المربع الحوارى التالى .....قم بتحديد AWT GUI Forms ثم أختر من file typeالعنصر Farme Formكما فى الشكل التالى :



_سيظهر لك مربع حوارى أخر لوضع اسم للنافذة الجديدة قم بفعل ذلك ثم اضغط على زر إنهاء المعالج (finsh)وهكذا نكون قد انشئنا نافذة كلاسكية تابعة للحزمة awt وطبعا تلك النافذة تكون اخف على النظام من النوافذ التابعة للحزمةswing وتكون بالشكل التالى :



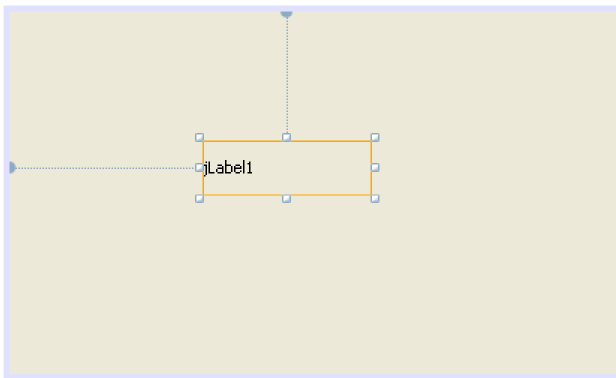
وطبعا تستطيع تغير خلفية النافذة من خلال الخاصية baceground ثم تقوم بأختيار اللون المناسب ....كما ان معظم الخصائص و الاحداث التى سبق و شرحناها تنطبق على تلك النوافذ....

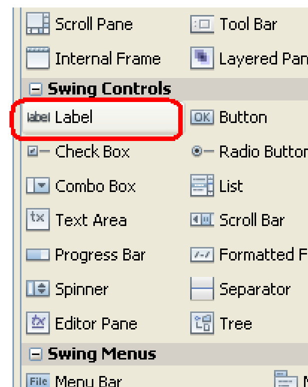
الفصل الثانى : حقل التسمية JLabel و زر الامر JButton

فى الدرس السابق تعلمنا وتحدثنا عن الاطارات و النوافذ و اليوم ان شاء الله تعالى سنتعلم كيفية التعامل مع حقول التسمية و زر الامر

أولا : العنصر الرسومى JLabel

يوفر العنصر الرسومى حقل التسمية JLabel تعليمات نصية أو معلومات فى فى واجهة المستخدم الرسومية و ذلك عن طريق أظهار سطر نصى أو صورة أو كليهما للقراءة فقط ولا يمكنك التعديل علية اثناء التشغيل .....ولكى تتمكنك من إضافة هذا العنصر لمشروعك كل ما عليك فعله هو الضغط على ايقونة العنصر فى نافذة الادوات ثم بعد ذلك وضع المؤشر على المنطقة المراد وضعه في نافذة برنامجك كما فى الصورة التالية:





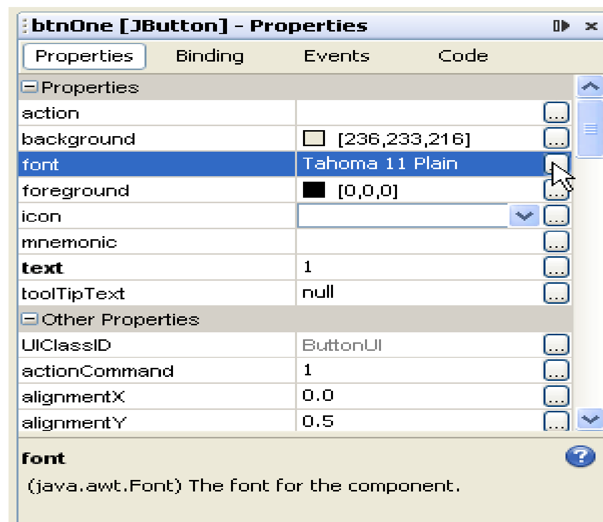
أهم خصائص العنصر الرسومى JLabel

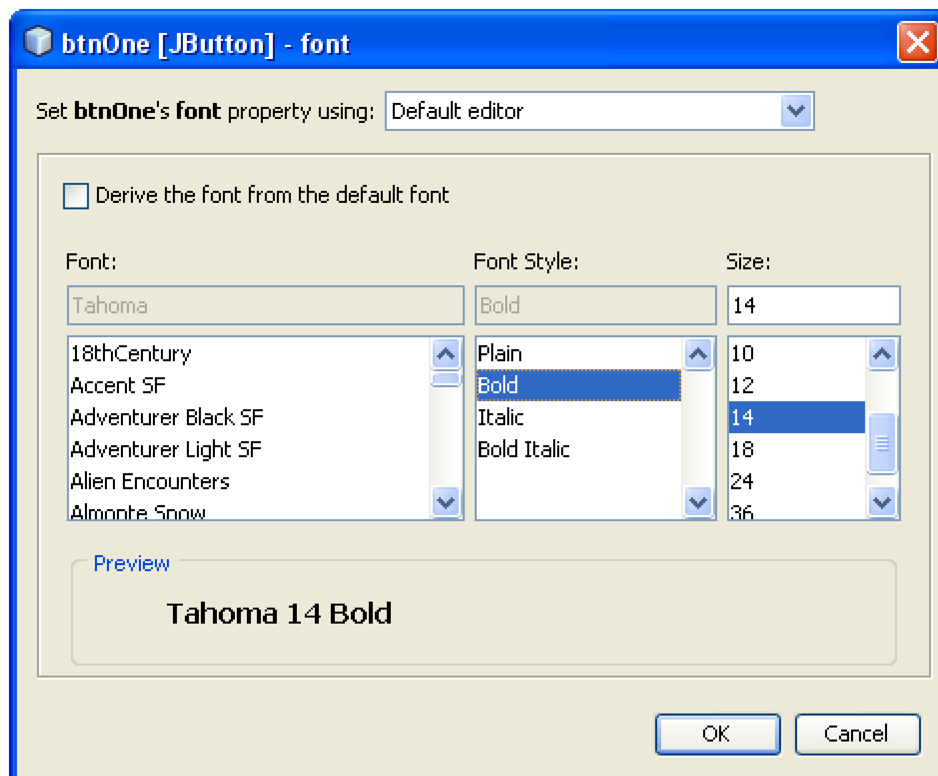
يوجد الكثير من الخصائص لحقل التسمية لكنى سأذكر فقط أهم الخصائص شائعة الاستخدام لهذا العنصر

1_الخاصية variable name والتى نستطيع من خلالها وضع أسم للعنصر للتعامل معه فى صفحة الكود

2_الخاصية variable Modifiers و التى من خلالها جعل العنصر عام أو خاص

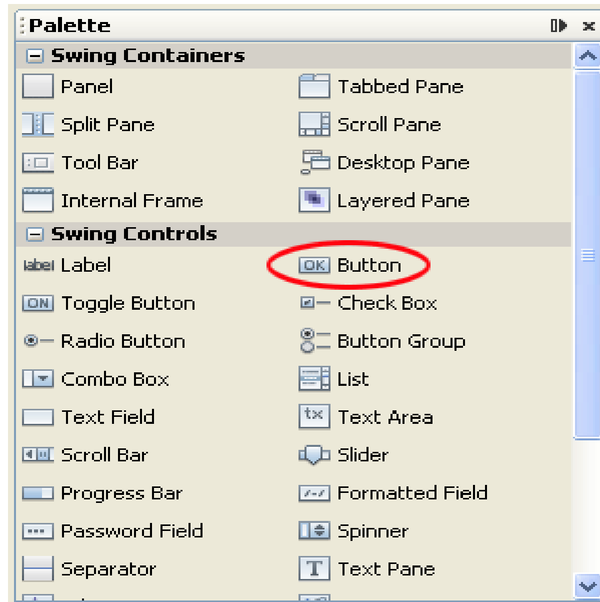
3_الخاصية font والتى من خلالها نستطيع تحديد الخصائص المميز للخط الحجم و النوع

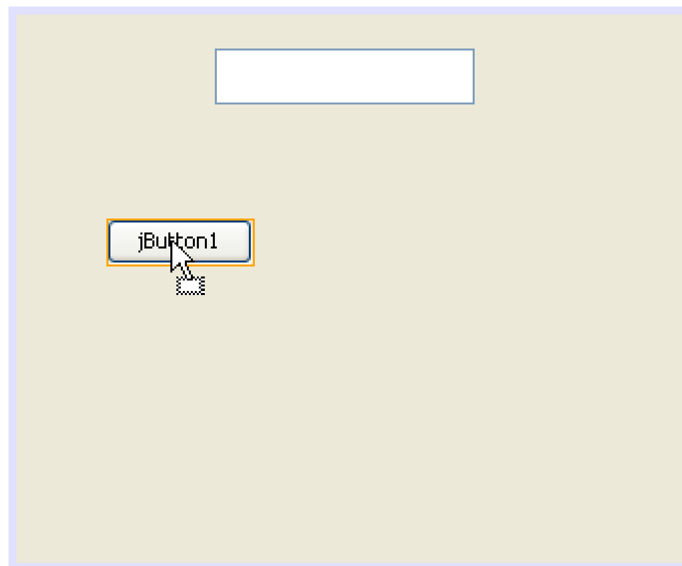




4_الخاصية foregraoud وهى المسؤلة لاعطاء لون معين للخط
5_الخاصية text و التى تضع اسم للعنصر يكون ظاهرا للمستخدم
6_الخاصية icon والتى من خلالها يتم وضع صورة مصغرة للعنصر

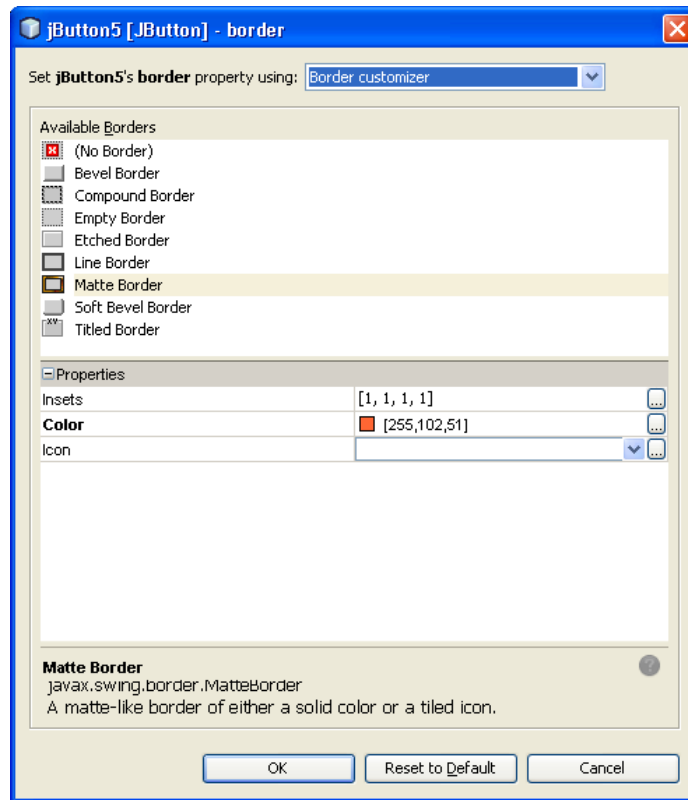
ثانيا : العنصر الرسومى JButton زر الامر





زر الامر هو عنصر رسومى يضغطة المستخدم لتنفيذ كود معين ....ومن اهم خصائص هذا العنصر الرسومى مايلى:

1_الخاصية variable name والتى نستطيع من خلالها وضع أسم للعنصر للتعامل معه فى صفحة الكود
2_الخاصية border والتى من خلالها نستطيع وضع إطار جميع للزر كما فى الشكل التالى

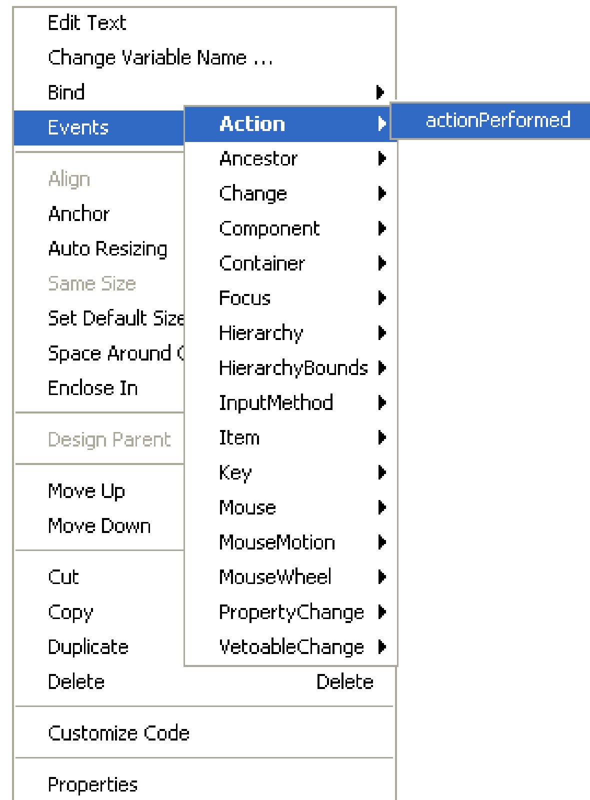


3_الخاصية rollover icon و التى تعطى صورة عند مرور الماوس على الزر
4_الخاصية icon والتى من خلالها يتم وضع صورة مصغرة للعنصر
5_الخاصية Horizontal Size و Vertical Size و اللتان تقوم بتحديد حجم الزر عن طريق وضع القيمة بها
و هناك ايضا الخاصية icon و text وfont و foregraoud

اهم أحداث العنصر الرسومى: يوجد الكثير من الاحداث لزر الامر ومنها مثلا...

أحداث الضغط على زر الامرKeyboard events
أحداث حركة الفارة mouse move events
أحداث الضغط على الفارة mouse click events
احداث الضغط على الزر Button click events

ويتم الوصول لتلك الاحداث كما فى الصورة التالية



وهو ان يتم تحديد زر الامر المراد تنفيذ كود فى حدث معين ثم كليك شمال فتظهر لنا قائمة الاحداث كما فى الصورة السابقة و ينتقل بينا المؤشر من صفحة التصميم الى صفحة كتابة الشفر....وفى المثال السابق قمنا بأختيار الحدث actionPerformed وهو يساوى ايضا الحدث.... MouseClicked سندخل صفحة الشفرة و نكتب الكود التالى:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
JOptionPane.showMessageDialog(null, "أهلا وسهلا");
}

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
}
```

الفصل الثالث : مكونات النصوص Text Components

اليوم إن شاء الله تعالى سنتعلم سويا كيفية كيفية التعامل مع مكونات النصوص فى الجافا **Text Components** والتى تشمل

JTextField, JTextArea ,JPasswordField, JFormattedTextField,

1 _العنصر الرسومى: JTextField

هذا العنصر الرسومى عبارة عن مربع لعرض و تحرير سطر نصى واحد ....ومن اهم الخصائص التى يمتيز بها هذا العنصر الرسومى ما يلى:

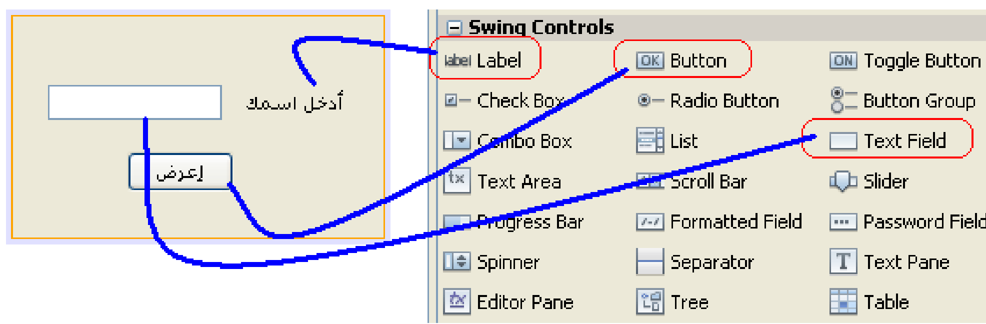
- الخاصية... editable و التى تجعل العنصر الرسومى لة القدرة على التحرير اثناء التشغيل من عدمه
- الخاصية... selectioncolor و التى من خلالها تستطيع تحيد اللون المخصص عند تحيد النص
- الخاصية ... selectextcolor-والتى من خلالها ايضا وضع لون معين للخط عند تحديد النص.

هذا بالاضافة للخصائص الرئسية الاخرى والتى تم شرحها سابقا فى العناصر الرسومية الاخرى ...مثل الخط و حجم الخط و لون الخط ولون الخلفية و غيرها.

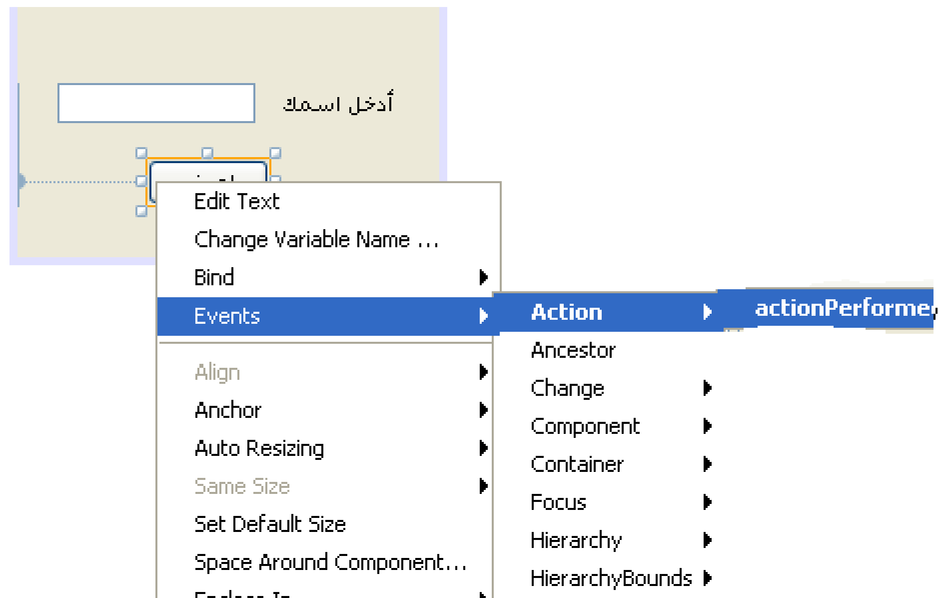
مثال : قم بإنشاء برنامج يقوم المستخدم بكتابة اسمه فى مربع النص و عند الضغط على زر (اعرض) يتم إظهار رسالة ترحيب للمستخدم...

الحل:

سنقوم بانشاء نافذة ونضع عليها عدد واحد حقل تسمية JLabel وعدد واحد زر امر jButton و عدد واحد حقل نصى من نوع JTextField كما فى الشكل التالى:



سنقوم بإنشاء الحدث ActionPerformed لزر الامر كما فى الشكل التالى:



تحت هذا الحدث نكتب الكود التالى:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

String name=jTextField1.getText();
 JOptionPane.showMessageDialog(null, " :"+ name);
}
```

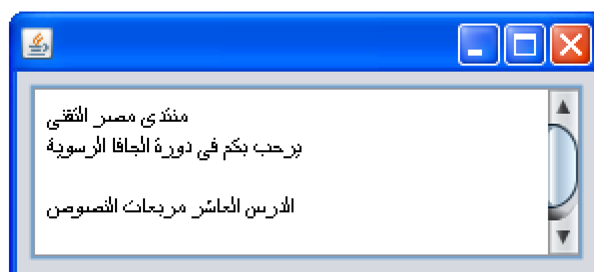
والنتيجة عند تشغيل المشروع تظهر الشاشة التالية و التى تطالبك بوضع اسمك و عند الضغط على زر الامر تظهر رسالة ترحيب بأسمك ...كما فى الاشكال التالية:
:



و بالتالى فإن الدالة **getText** هى التى من خلالها الحصول على القيمة الموجودة فى مربع النص TextField

2 _ العنصر الرسومى ... JTextArea

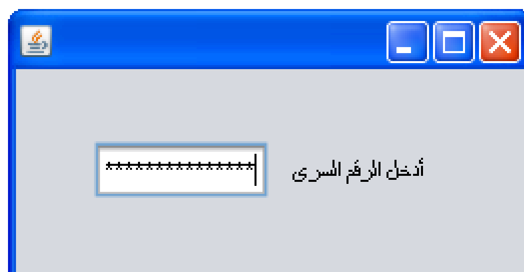
هذا العنصر الرسومى عبارة عن مربع ايضا لعرض وتحرير عدة اسطر و ليس سطر واحد مثل JTextField لذلك اذا اردت ان يقوم مستخدم برنامج بكتابة عدة اسطر نصية لذلك يجب عليك استخدام هذا العنصر الرسومى ....و يتميز هذا العنصر الرسومى بكافة الخصائص السابقة بالاضافة الى خاصية شريط التمير الافقى و الراسى ..كما فى الصورة التالية:

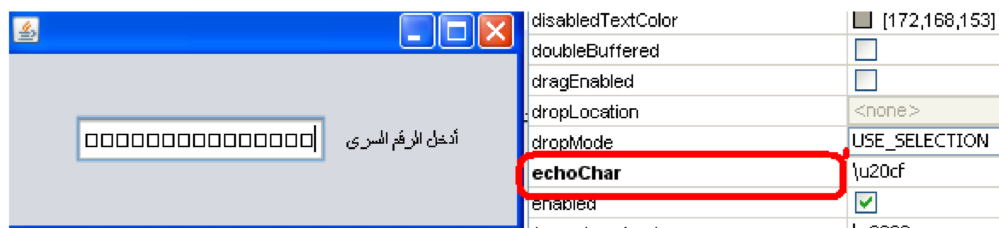


3 _العنصر الرسومىJPasswordField

هذا العنصر الرسومى عبارة عن مربع نص لعرض وتحرير المدخلات باشكال و رموز اخرى عندما يقوم المستخدم بكتابة المدخلات فية و الرمز الافتراضى لتلك البيانات هى علامة النجمة (*).ويستخدم هذا العنصر الرسومى فى كتابة الارقام السرية لبرنامج مثلا ...

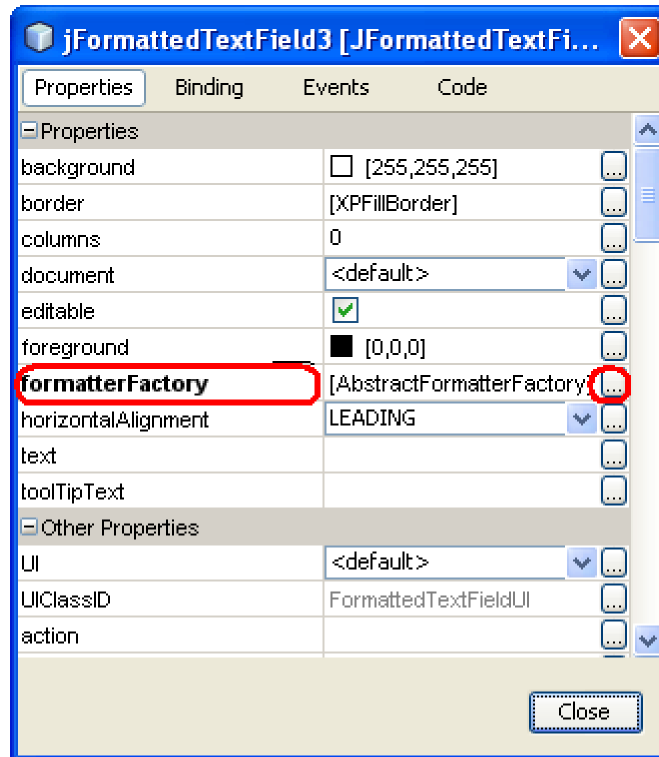
.ومن اهم الخصائص المميزة لهذا العنصر هى خاصية EchoChar و التى يمكن من خلالها تغير الرموز و الاشكال المراد ظهورها بدلا من الرمز الافتراضى النجمة.(*)

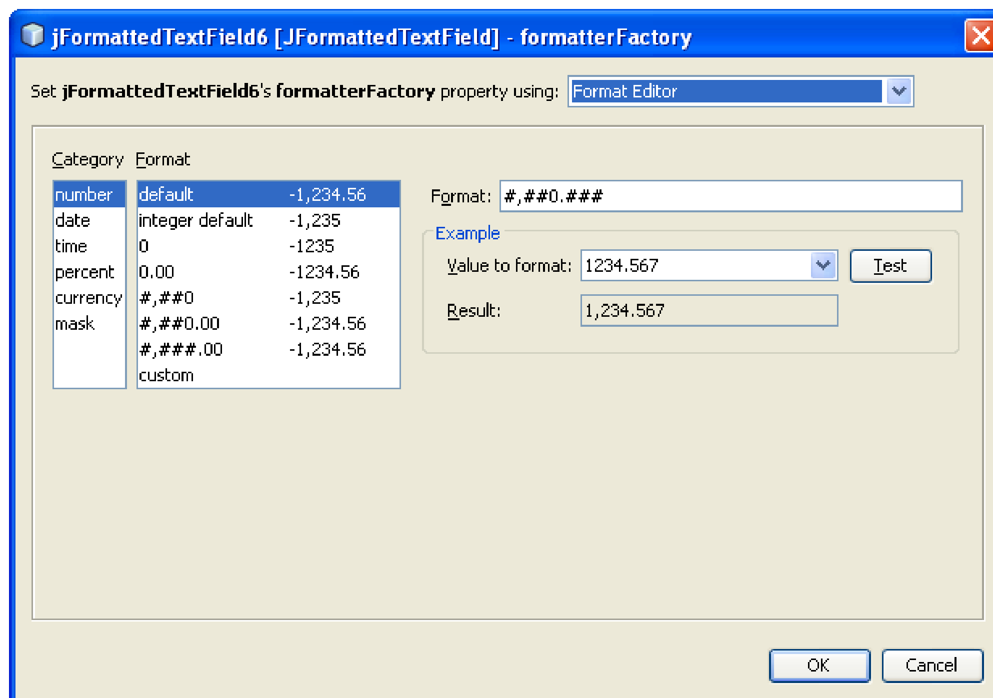




4 _ العنصر الرسومى JFormattedTextField

هذا العنصر الرسومى هو عبارة عن مربع نص لتحرير و عرض البيانات بتنسيقات مختلفة ...كأن تجعله يقوم بعرض و تحرير ارقام فقط او تواريخ او اى صيغة اخرى انت تحددها ...و يمكن فعل ذلك عن طريق الخاصية formatterFactory و التى تهظر لك مربع حوارى لاستخدام التنسيقات المختلفة كما فى الشكل التالى:





والنتجية....

just letter (mask) abcde

just number (mask) 12345

with a format (mask) (1234) 56789

date 20-11-2011

number with decimal 1.000,00

percent 10,00%

number 40.000

result 44.000

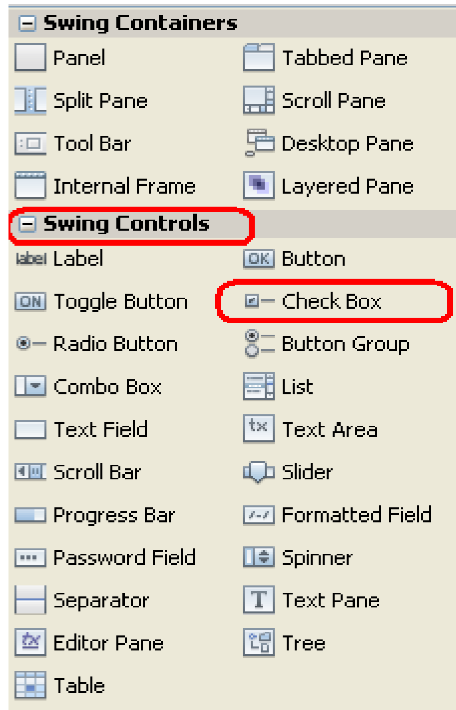
calculate

الفصل الرابع : مربع الاختيار المتعدد jCheckBox و خانة الاختيار jRadioButton

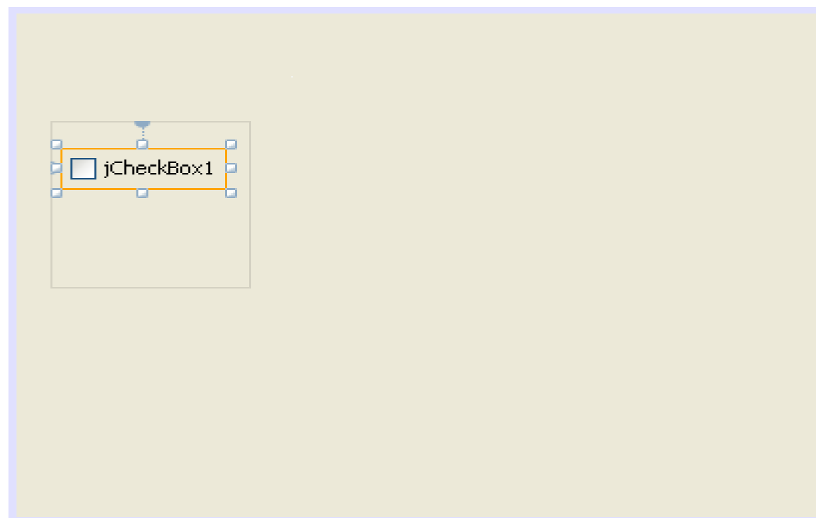
فى هذا الدرس إن شاء الله نواصل معرفة المزيد عن واجهة المستخدم الرسومية فى الجافا و سنتعلم كيفية التعامل مع عنصرين مهمين هما مربع الاختيار المتعدد jCheckBox و خانة الاختيار jRadioButton

أولا : مربع الاختيار المتعددjCheckBox

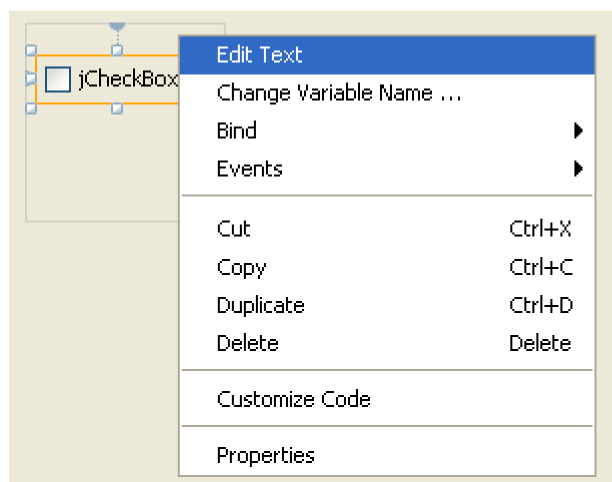
هو عبارة عن عنصر تحكم يسمح للمستخدمين اختيار العناصر عن طريق تحديد أو إلغاء تحديد مربع التجزئة الذى يوفيرها في الاختيار لاختيار أكثر من عنصر فى وقت واحد ...و يمكنك الحصول على مربع الاختيار المتعدد من خلال قائمة Swing Containers كما فى الشكل التالى:



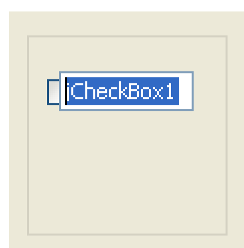
الان قم بسحب العنصر بالماوس ثم اسقطه على نافذة مشروعك كما فى الشكل التالى:



وكما تلاحظون شكل مربع الاختيار المتعدد على نافذة المشروع يأخذ النص jCheckBox1 هو النص الافتراضي للعنصر.و يمكنك تغيير هذا النص إما من خلال نافذة الخصائص ، أو بالنقر بزر الماوس الأيمن على مربع الاختيار. و من القائمة التي تظهر، حدد تحرير النص في الصورة أدناه



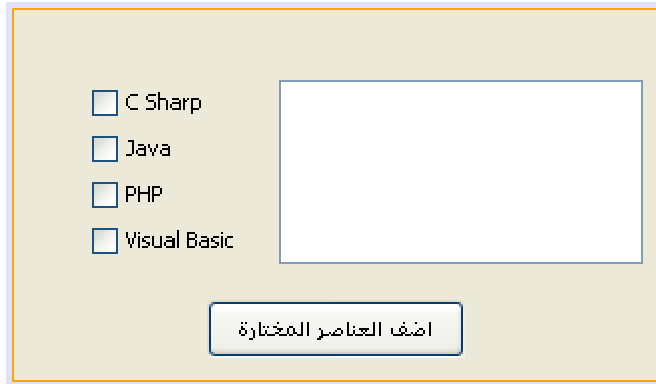
عند النقر على تحرير النص، سيتم تسليط الضوء على النص الافتراضي كما فى الشكل التالى:



وهكذا نستطيع تغير النص للعنصر وليكن C Sharp كما يلى:



سنقوم بعد ذلك بإضافة العنصرJTextArea على النموذج بحيث عندما يقوم المستخدم بتحديد لغة أو لغات وعند الضغط على زر الامر (اضف) سوف يتم سردها في منطقة النص...JTextArea اذن الشكل النهائى للنافذة ستكون كما فى الصورة التالية:

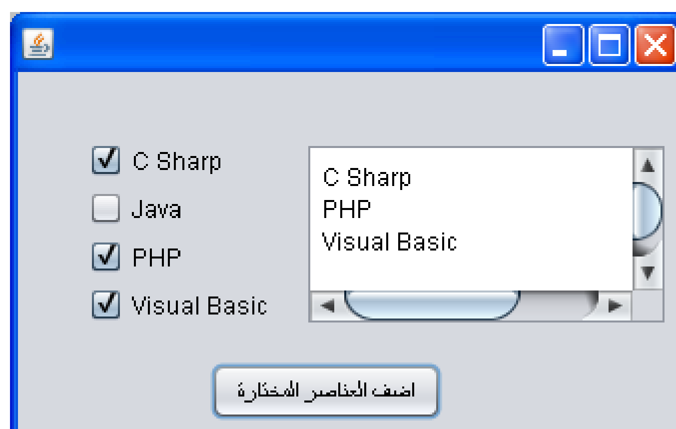


نحن الان بحاجة لمعرفة أي خانة الاختيار تم اختيارها من قبل المستخدم حتى نتمكن من إضافتها فى العنصر الرسومى..... jTextAreaوفى مربع الاختيار يحتوي على خاصية تسمى () isSelected التي ترجع حقيقية منطقية لمعرفة إذا تم تحديد خانة الاختيار أم لا ... سنقوم ايضا باستخدام جملة IF و ذلك على النحو التالى:

الرمز:

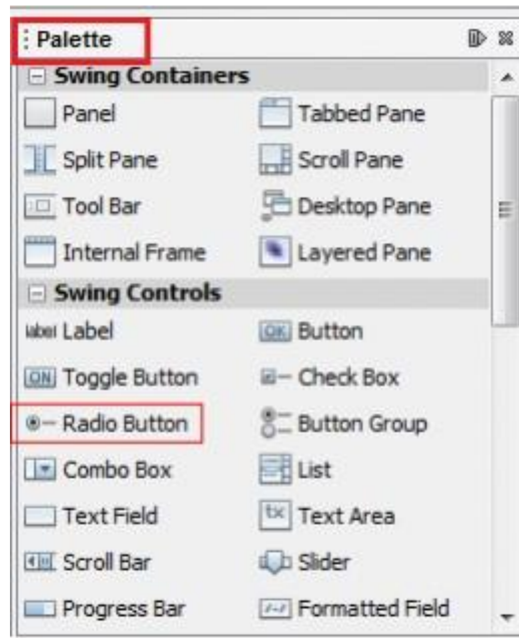
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String s="" ;
if (jCheckBox1.isSelected()){
   s=s+""+jCheckBox1.getText()+'\n';
}
if (jCheckBox2.isSelected()){
    s=s+""+jCheckBox2.getText()+'\n';
}
if (jCheckBox3.isSelected()){
    s=s+""+jCheckBox3.getText()+'\n';
}
if (jCheckBox4.isSelected()){
    s=s+""+jCheckBox4.getText()+'\n';
}
jTextArea1.setText(s);
}
```

وكما ترون فى الكود السابق قمنا بالاعلان عن المتغير s فإذا تم تحديد خانة الاختيار نحصل على نص التسمية التوضيحية من مربع الاختيار. ثم يتم تخزين هذا النص في المتغر s جبنب الى جنب مع وضع حرف n/ و ذلك لوضع المؤشر فى سطر جديد ....ثم استخدمنا الدالة setText للعنصر الرسومى jTextArea1 و ذلك لوضع قيمة المتغير s فى العنصر الرسومى ....قم الان بتشغيل برنامج و ستجد ما يلى:

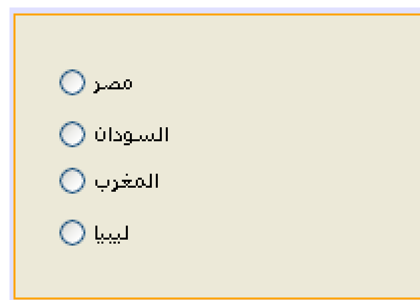


ثانيا : خانة الاختيار jRadioButton

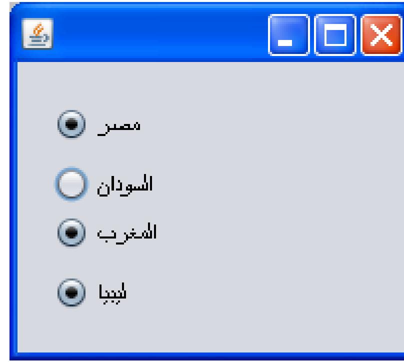
خانة الاختيار هى احد العناصر الرسومية و التى تسمح لمستخدم باختيار عنصر واحد فقط على عكس مربع الاختيار المتعدد الذى تم شرحه اعلاه .....ويمكن الحصول على خانة الاختيار jRadioButton من قائمة Swing Containers كما بالشكل التالى:



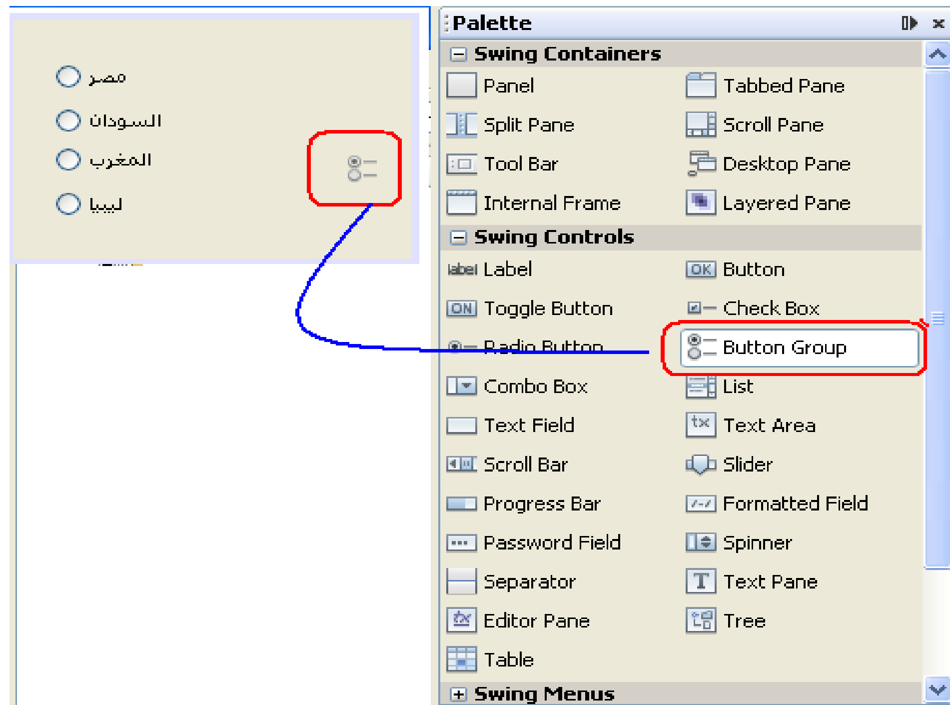
و يمكن تحرير النص الافتراضى لتسمية jRadioButton كما فعلنا مع خانات الاختيار أعلاه ....قم باضافة عدد من خانات الاختيار وقم بتغير التسمية الافتراضية كما فى الشكل التالى:



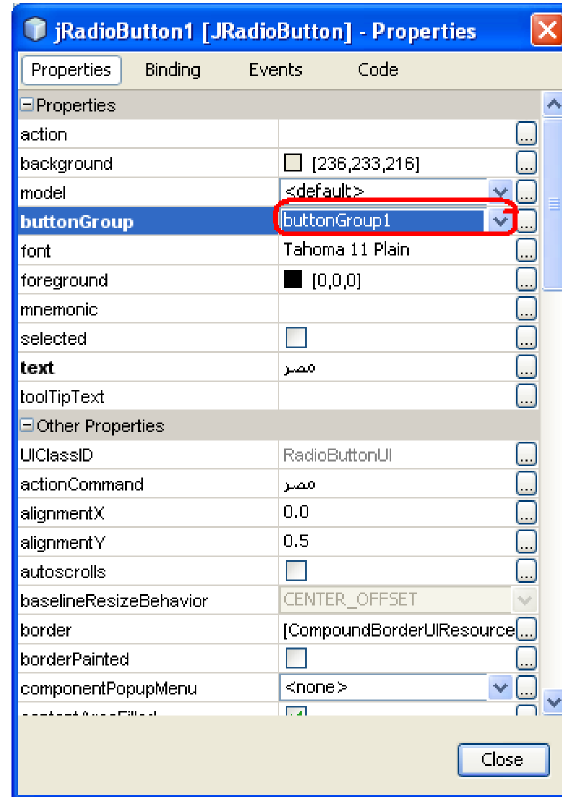
وعند تشغيلك لمشروعك ستجد أن خانة الاختيار jRadioButton لا تعمل بشكل صحيح ...أذن انها من المفترض أن تجعل المستخدم غير قادر على أختيار إلا عنصر واحد فقط من تلك العناصر ...لكننا لو قمنا بشتغيل مشروعنا الان سنجد ان المستخدم قادر على اختيار اكثر من عنصر كما فى الشكل التالى:



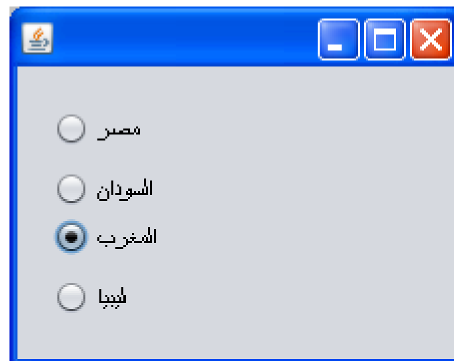
اذن ولحل تلك المشكلة ما عليك سوى الرجوع الى قائمة Swing Containers ستجد المكون ButtonGroup قم بسحبه بالماوس و ضعه على النافذة ....كما فى الشكل التالى:



بعد اسقاط العنصر الرسومى ButtonGroup على النافذة ستجد أنه لم يعد ظاهرا عليها فلا تقلق من ذلك ....الان قم بتحديد خانة الاختيار jRadioButton ومن نافذة الخصائص ستجد الخاصية ButtonGroup ومن خلالها ستحدد المجموعة التى تريد أن ينتمى اليه خانة الاختيار هذه ....وذلك كما فى الشكل التالى:



كرر تلك العملية لجميع خانات الاختيار فى مشروعك و بعدها شغل برنامج ستجد أنه يعمل بكفاءة و لن يستطيع المستخدم حينها اختيار اكثر من عنصر كما فى الشكل التالى:



وللحصول على القيمة التى تم اختيارها من خلال خانة الاختيار تلك نستخدم الدالة أو الطريقة isSelected كما فى الكود التالى:

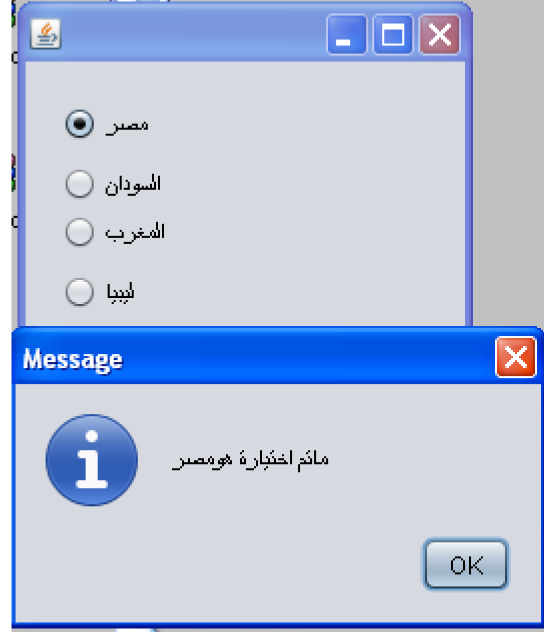
الرمز:

```
private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {

   if(jRadioButton1.isSelected()){
   String textradio = jRadioButton1.getText();

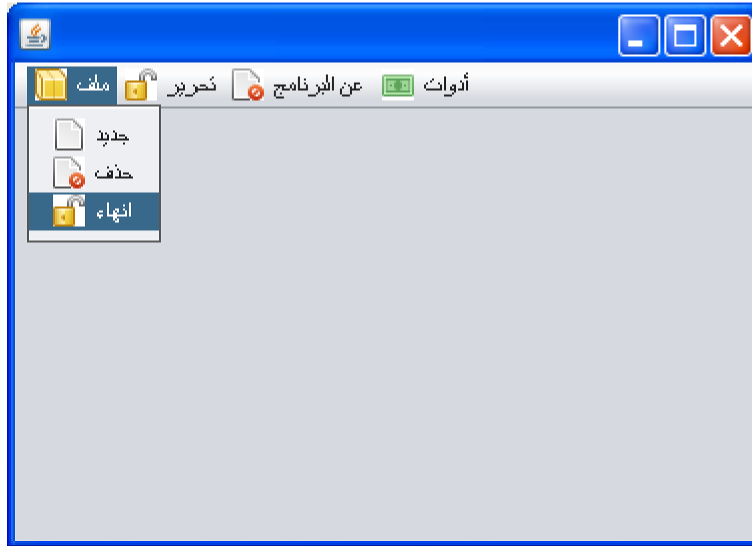
   JOptionPane.showMessageDialog(null, "" + textradio);
}
}
```

و النتجية كما فى الشكل التالى:



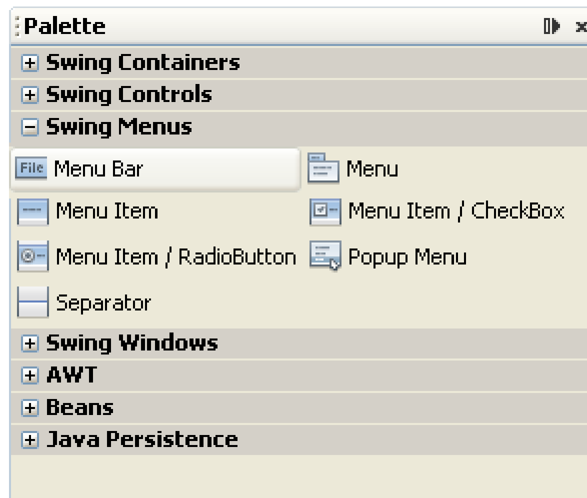
الفصل الخامس : اشرطة القوائم Menu Bar

ومازلنا نتحدث عن واجهة المستخدم الرسومية.. اليوم سنتحدث عن العنصر الرسومى (**اشرطة القوائم**..**Menu Bar**) و التى تمكنك من إضافة قوائم على نافذة مشروعك بحيث تسمح للمستخدم باختيار من عدة عناصر وتشمل مثلا قائمة ملف قائمة تحرير ،....الخ غير ذلك كما فى الصورة التالية:

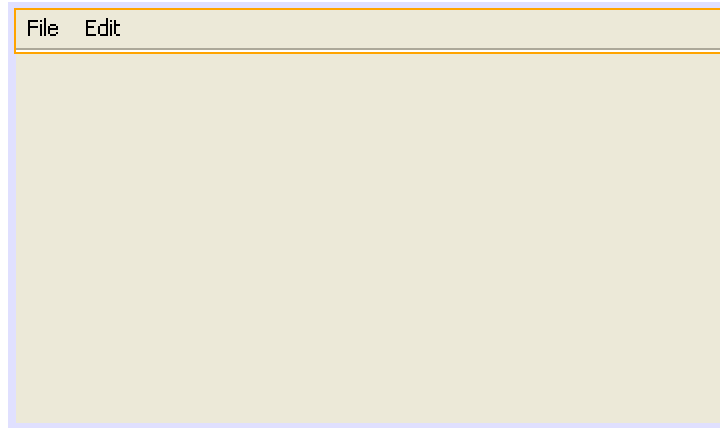


ولكى يتم الحصول على الشكل أعلاه أتبع الخطوات التالية:

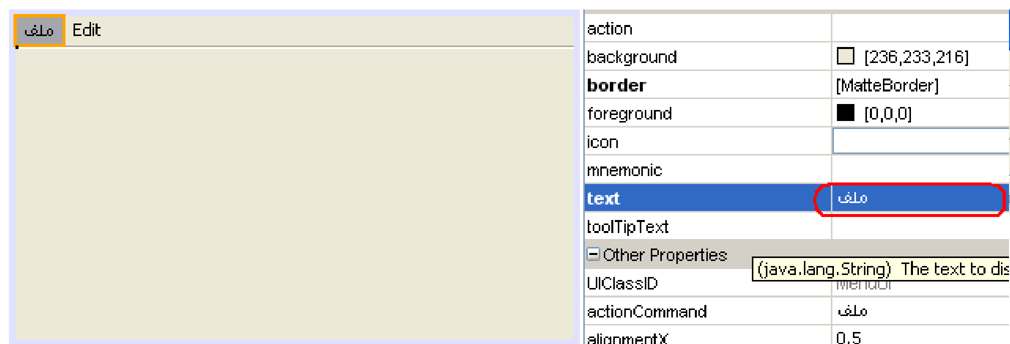
1_ قم بإضافة Menu Bar الى نافذة مشروعك و ذلك بسحب العنصر من نافذة palette تحت التصنيف Swing controlsثم اسقاطة على نافذة المشروع كما فى الشكل التالى:



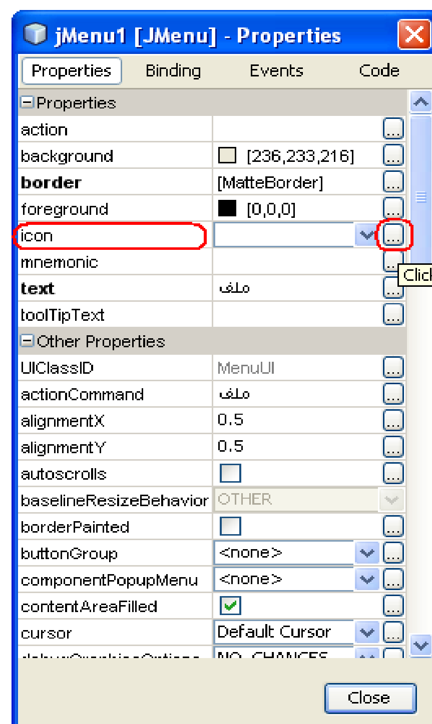
وعند اسقاط العنصر الرسومى Menu Bar على نافذة مشروعك ستجد أن العنصر يحتوى على قائمة ملف و تحرير كما فى الشكل التالى:



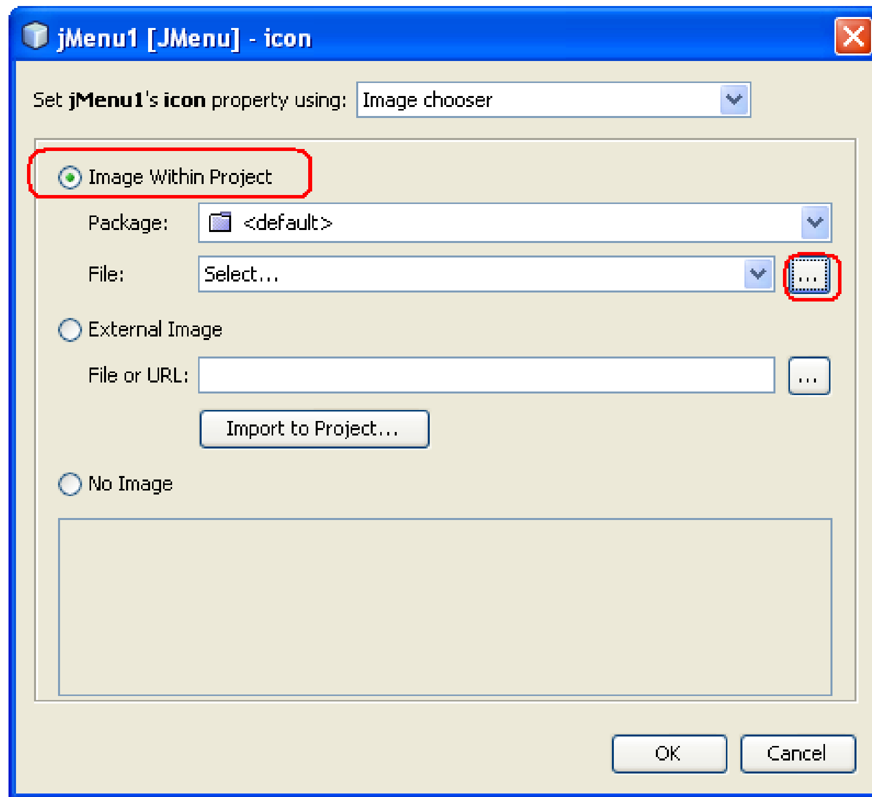
2_ولكى تتمكن من تغير النصوص الافتراضية لقوائم العنصر الرسومى Menu Bar الى الاسماء التى تريدها قم بتحديد القائمة file ثم من نافذة الخصائص حدد الخاصية text و قم بتغير مسمى القائمة كما فى الشكل التالى:



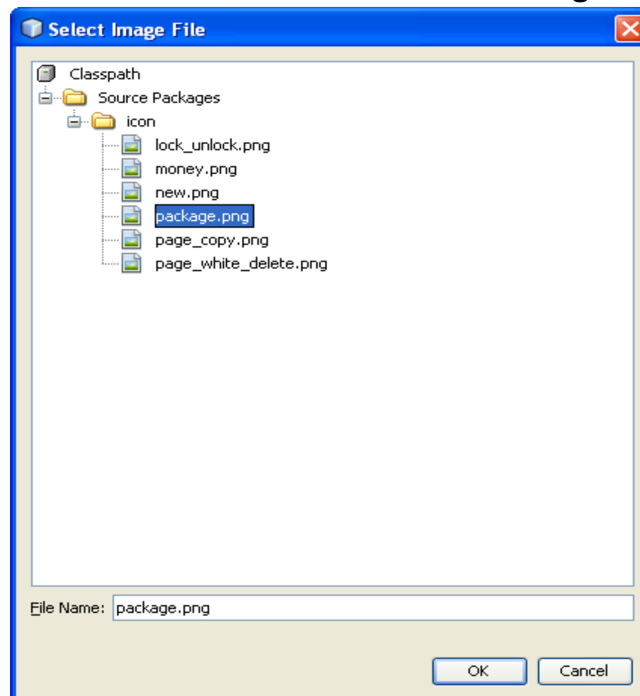
3_يمكنك ايضا إضافة ايقونة للقائمة ملف.... فقط قم بتحديد القائمة ومن نافذة الخصائص حدد الخاصية icons كما فى الصورة التالية:



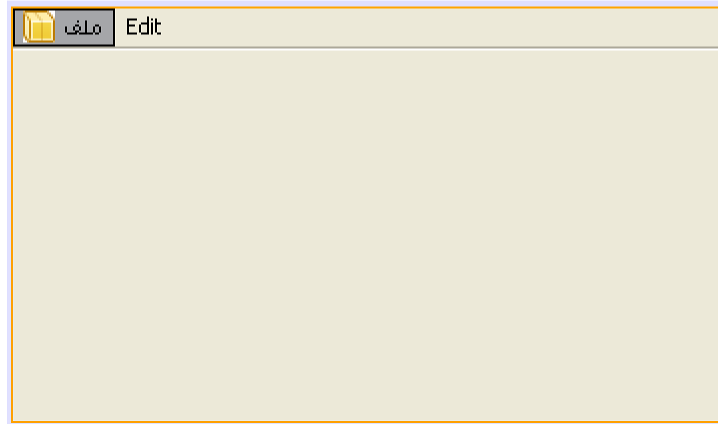
من الشكل أعلاه قم بالضغط على الزر الصغير الذى بالجانب الايمن من الخاصية يتم اظهار المربع الحوارى التالى:



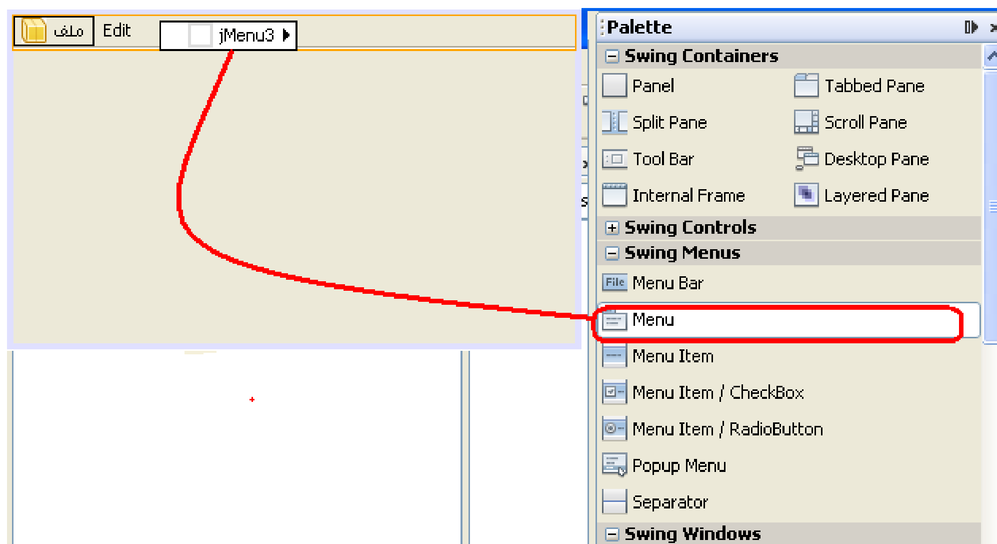
هذا المربع الحوارى يتيح لك كيفية ادراج صورة للعنصر اما من خلال مجلد صور مرفق مع البرنامج او عن طريق مجلد اخر او حتى رابط انترنت ......لذلك و من الافضل لنا هو ادراج مجلد صور به جميع ايقونات وصور البرنامج و ذلك ليتم ارفاقة مع المشروع اينما ذهب ....و لكيفية عمل ذلك ما عليك فقط الا ادراج هذا المجلد داخل مجلد البرنامج المسمى src .... فعلى سبيل المثال قمت بعمل مجلد يسمى icon ثم وضعته داخل المجلد src الخاص بمشروعى و بالتالى عند الضغط على زر استعراض كما فى الشكل السابق سأجد مجلد الصور الخاصى بى و سأختار منه الصورة التى اريدها للقائمة كما فى الشكل التالى:



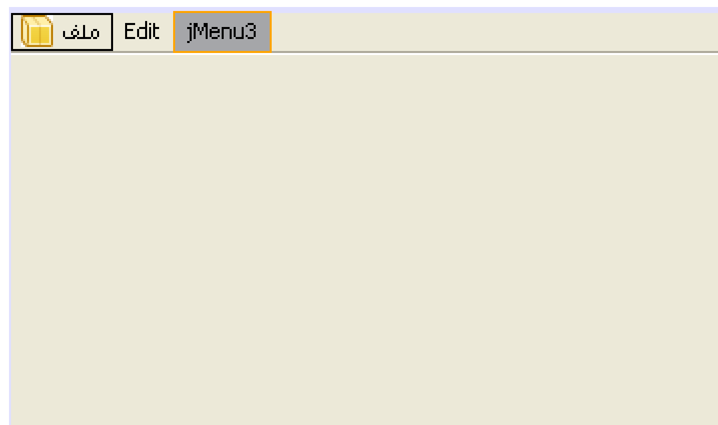
و النتيجة كما يلى:



4_اضافة قوائم أخرى.... لكى تتمكن من إضافة اى قائمة اخرى لشريط Menu Bar ما عليك اللى التركيز على قائمة الادوات palette ثم تقوم بسحب العنصر الرسومى menu ووضعة على شريط القوائم كما فى الشكل التالى:



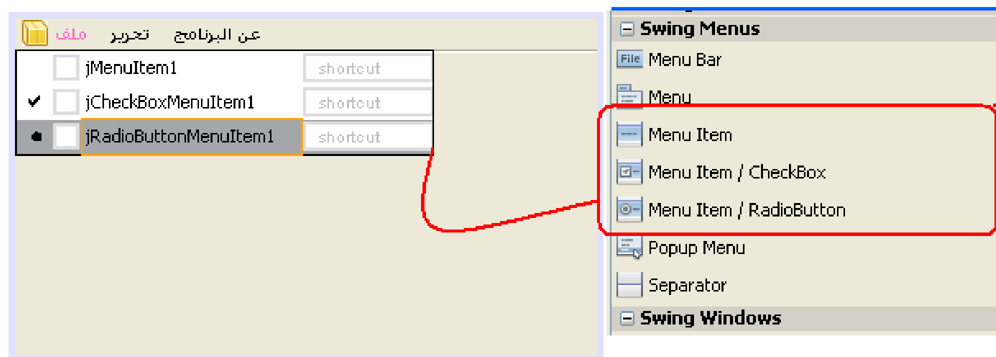
ليصبح بالشكل التالى بعد اسقاطة على نافذة مشروعك و يمكنك الان تغير اسمه ووضع ايكون له كما فعلنا أعلاه..



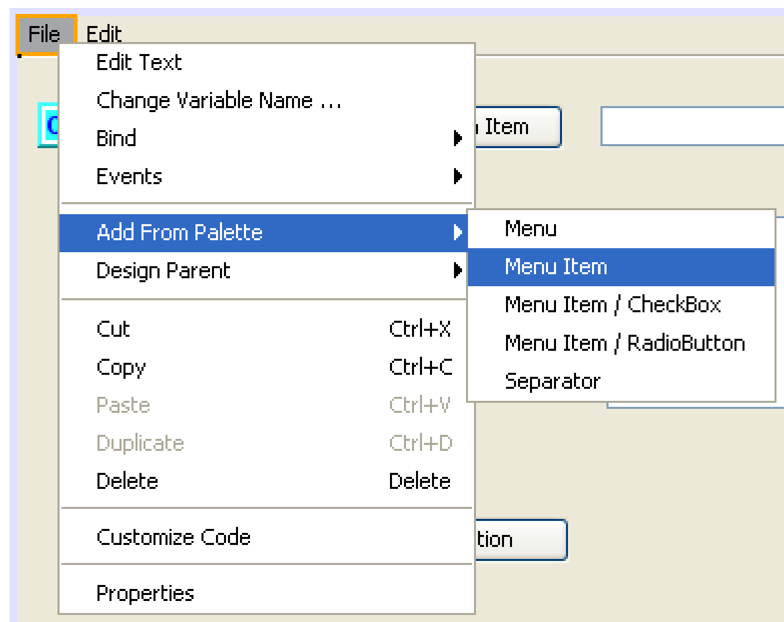
5_كيفية إضافة عناصر للقوائم.... يوجد ثلاث انواع من العناصر التى تستطيع إضافتها لاى قائمة وذلك على النحو التالى:

ا _العنصر... Menu Item هو عنصر رسومى يحتوى على بيانات عند الضغط علية القائمة الرئسية يتم اظهاره
ب _ العنصر... CheckBoxMenuItem هو نفس العنصر السابق ولكنه يشمل علامة (صح)عند اختيار المستخدم له
ج_العنصر... RadioButtonMenuItem وهو يشبة العناصر السابقة ولكنه يشمل علامة (٠) عند اختيار المستخدم له

والان تعالوا نتعلم كيفية اضافة تلك العناصر لاى قائمة ...يتم ذلك اما من خلال النافذة palette وذلك بسحب تلك العناصر و اسقاطها تحت القائمة المراد ان تظهر بها كما فى الشكل التالى:



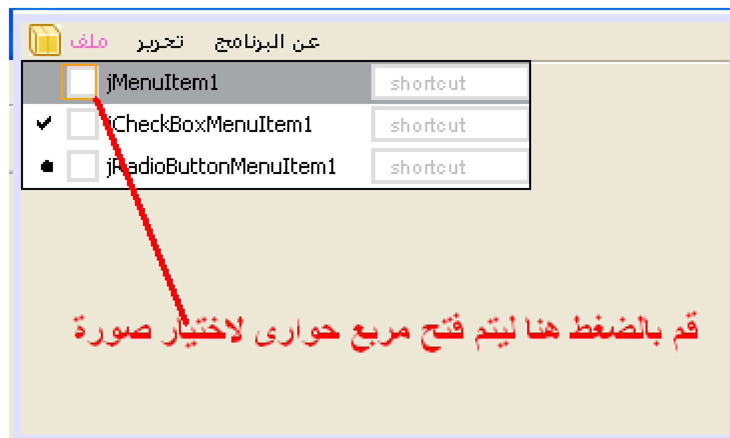
او من خلال تحديد القائمة المعنية ثم كليك شمال ثم نختار الامر Add From Palette ثم نختار العنصر المراد وضعة للقائمة كما فى الشكل التالى:



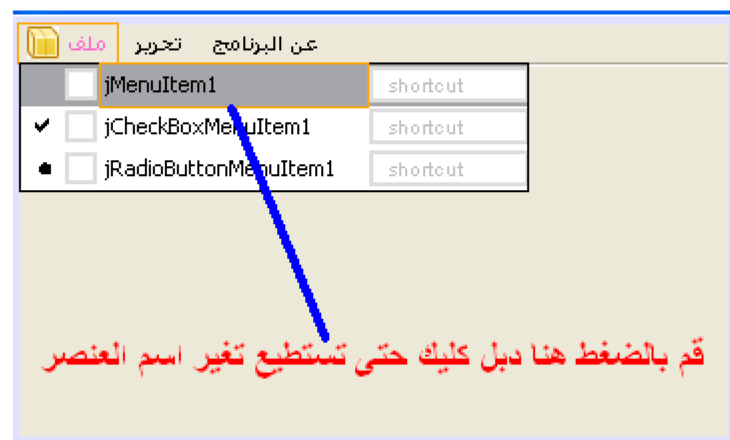
واى عنصر من العناصر السابقة له ثلاث مواضع مهمة.....

ا_ الموضع icon لوضع صورة للعنصر الفرعى ...قم بتحديد العنصر ثم اضغط على موضع الايقونة دبل كليك ليتم فتح

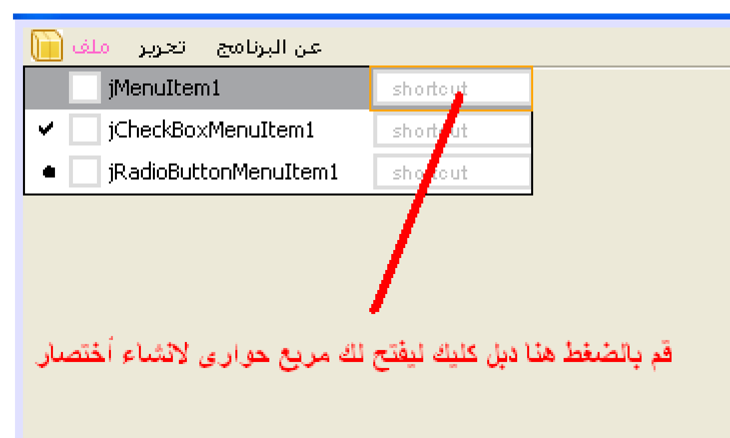
لك المربع الحوارى لاختيار الصورة كما فى الصورة التالية:



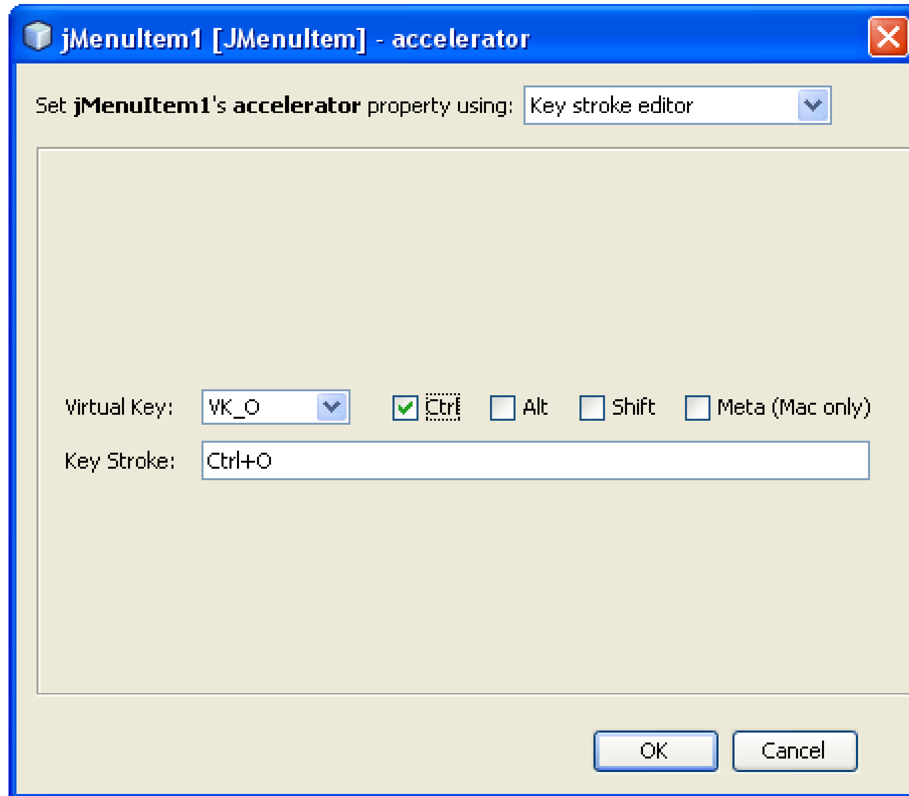
ب _ موضع تغير اسم العنصر ....قم بالضغط عليه دبل كليك لتستطيع تغير اسم العنصر كما فى الشكل التالى:



ج _ الموضع shortcut و الذى من خلالة تستطيع انشاء اختصار لهذا العنصر قم بالضغط كما فى الشكل التالى:



ليظهر لك المربع الحوارى التالى:



كما هو ظاهر اعلاه ... يمكنك تعيين مفاتيح الاختصار الذي تريد لعنصر القائمة من هذا المربع الحوار ....فمثلا اردنا ان يكون مفتاح الاختصار لهذا العنصر هو... CTRL + O فقمنا بتنشيط خانة CTRL ثم أخترنا القيمة vk_o ثم اوكية ...كما هو ظاهر اعلاه.

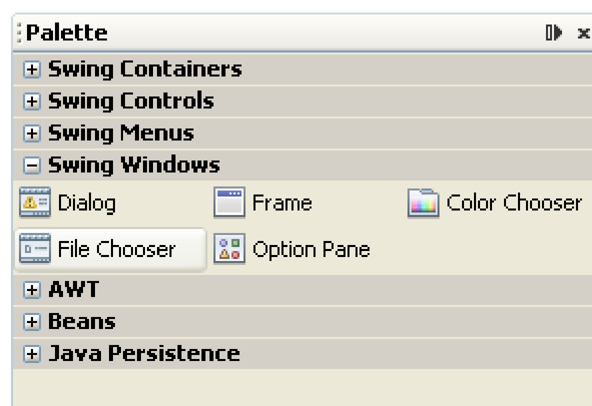
ويمكنك تنفيذ اى امر تحت اى عنصر من خلال Events > Action > Action Performed

الفصل السادس : التعامل مع المربع الحوارى jFileChooser

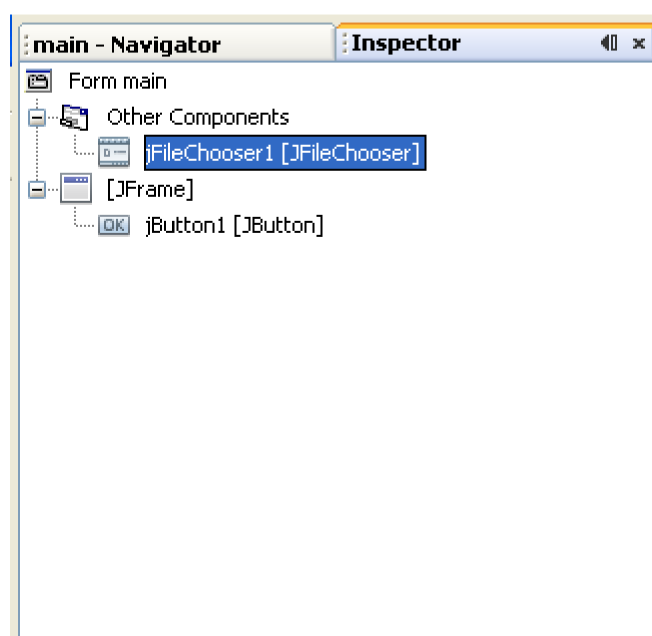
اليوم سنتعلم سويا الدرس الرابع عشر من دورة الجافا الرسومية و هو عن كيفية التعامل مع مربع الحوار File Chooserو الذى يتم استخدامة فى فتح وحفظ الملفات

كيفية أستخدام مربع الحوارJFileChooser

المربع الحوارى JFileChooser هو أحد العناصر الرسومية التى توفرها الجافا التى تستخدم لاستعراض الملفات و المجلدات من على جهاز المستخدم ومن ثم فتح أو حفظ ملف من والى الجهاز أو حتى لانشاء مجلد.....ويمكنك الحصول على هذا العنصر الرسومى من قائمة palette كما فى الشكل التالى:



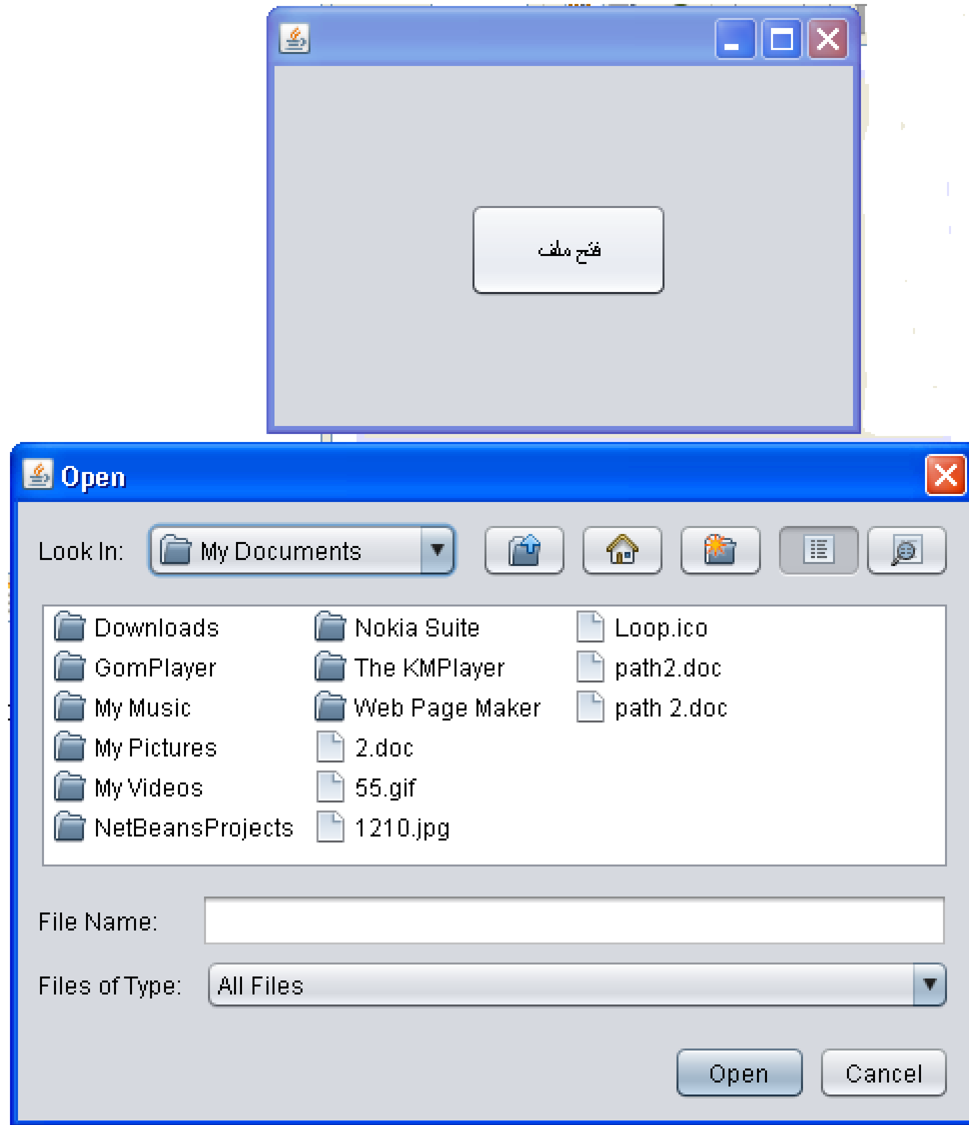
قم إسقاط العنصر الرسومى تحت نافذة مشروعك وليس على النافذة نفسها حتى لا يتم رسم العنصر عليها....أذن كل ما عليك فعله اسقاط العنصر المنقطة البياض اسفل النافذة و ستجدة كما فى الشكل التالى...



و لكى نتمكن من اظهار المربع الحوارى نقوم بوضح زر أمر على النافذة وفى حدث ActionPerformed نكتب الكود التالى:

```
jFileChooser1.showOpenDialog(this);
```

وهذا هو الكود الخاص باظهار المربع الحوار) فتح (وعند تشغيل مشروعك و الضغط على زر الامر فتح سيظهر لك المربع الحوارى كما فى الشكل التالى:



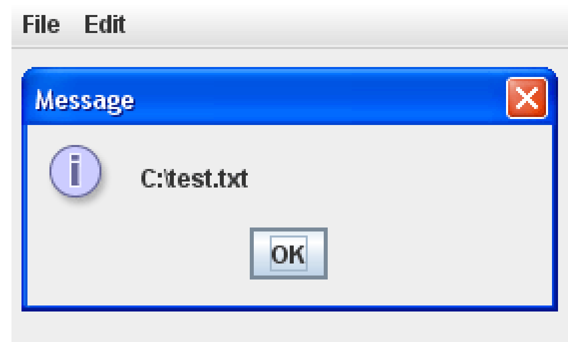
بهذا المربع الحوارى حتى الان لا نستطيع فتح أى ملف و وكل ما نستطيع فعله هو استعراض و التنقل بين الملفات و المجلدات ....اذن لكى نتمكن من فتح ملف معين من خلال هذا المربع الحوارى نستخدم الدالة getSelectedFile وذلك بتعديل الكود السابق ليصبح كالتالى:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

int egy =jFileChooser1.showOpenDialog(this);
if (egy==javax.swing.JFileChooser.APPROVE_OPTION){
 java.io.File file =jFileChooser1.getSelectedFile();
 String file_name = file.toString( );
 JOptionPane.showMessageDialog(null, file_name);
}
}
```

- فى السطر الثانى قمنا بتعريف متغير من نوع int يسمى egy ثم جعلناه يحمل مربع الحوار (فتح)
- السطر الثالث : استخدمنا جملة شرطية لمعرفة ما اذا كان المستخدم قام بالضغط على زر (موافق) فى المربع الحوارى (فتح) أم لا ...و استخدمنا لذلك APPROVE_OPTION و التى تعنى زر الموافق فى المربع الحوارى
- السطر الرابع ...اذا تحقق من الشرط يقوم بالحصول على اسم الملف المختار ...و استخدمنا لذلك الكائن file من الفئة **io** وبالتالي فإن الملف الذي اختاره المستخدم في نهاية المطاف يتم تخزينة في المتغير file
- السطر الخامس: قمنا بالاعلان عن المتغير file_name كمتغير نصى و بعد ذلك جعلناه يقوم بحفظ القيمة المخزنة فى الكائن file بعد تحويلة الى نص عن طريق الامر toString
- السطر السادس: ظهور رسالة بأسم الملف المختار

الان قم بتشغيل البرنامج ثم أنقر على زر الامر فتح ....لمشاهدة مربع الحوار (فتح). حدد أي ملف على جهاز الكمبيوتر الخاص بك وانقر فوق فتح...ستظهر لك رسالة بأسم الملف المختار كما فى الشك التالى:



تصفية الملفات المختارة

عند استخدامك للمربع الحوارى ستلاحظ أنه يتم تعيين أنواع الملفات إلى "جميع الملفات". ...و يمكنك تصفية الملفات في هذه القائمة، بحيث يمكن للمستخدم فتحها فقط، دون غيرها .....كالملفات النصية، أوصور ذات امتدادات معينة (JPEG، GIF، PNG). (نصوص، صور، ميديا.. الخ) و يتم ذلك باستخدام الطريقة addChoosableFileFilter بالاضافة الى الكائن FileFilter و الكائن FileNameExtensionFilter و ذلك على النحو التالى:

فى البداية قم بإستيراد الكائنات كما يلى:

```
import javax.swing.filechooser.FileFilter;
import javax.swing.filechooser.FileNameExtensionFilter;
```

ولتصفية الملفات نحتاج الى إنشاء كائن FileFilter جديد و ذلك كما يلى:

```
FileFilter ft = new FileNameExtensionFilter("Text Files", "txt");
```

و كما تلاحظون أننا قمنا بوضع نوع الملفات المراد اظهارها و امتداها بين قوسين("Text Files", "txt")
بعد ذلك نستخدم الدالة addChoosableFileFilter لتصفية الملفات كما يلى:

```
jFileChooser1.addChoosableFileFilter(ft);
```

و بالتالى فإن الكود الكامل سيصبح كما يلى:

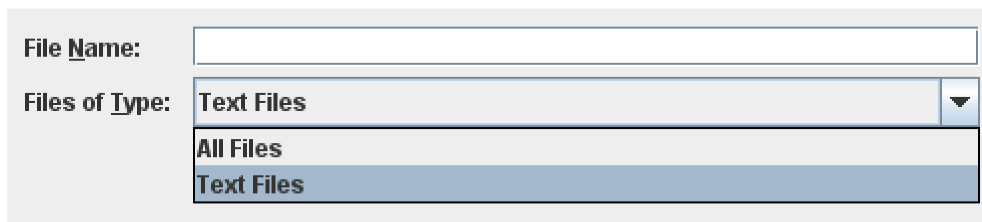
الرمز:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

FileFilter ft = new FileNameExtensionFilter("Text Files", "txt");
 jFileChooser1.addChoosableFileFilter(ft);

int egy =jFileChooser1.showOpenDialog(this);
if (egy==javax.swing.JFileChooser.APPROVE_OPTION){
 java.io.File file =jFileChooser1.getSelectedFile();
 String file_name = file.toString( );
 JOptionPane.showMessageDialog(null, file_name);
}
}
```

و النتجية



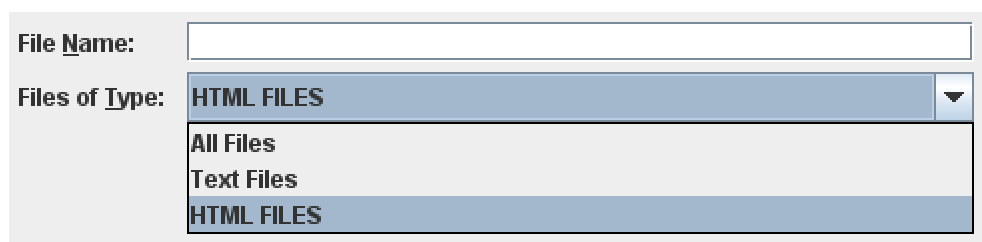
و إذا كنت تريد إضافة نوع أخر من الملفات في القائمة، (لعرض ملفاتHTML ، على سبيل المثال)، يمكنك إعداد كائن آخر:FileFilter

```
FileFilter ft = new FileNameExtensionFilter("Text Files",
"txt");
 FileFilter ft2 = new FileNameExtensionFilter("HTML Files",
"html");

 jFileChooser1.addChoosableFileFilter(ft);

jFileChooser1.addChoosableFileFilter(ft2);
```

و النتيجة ايضا



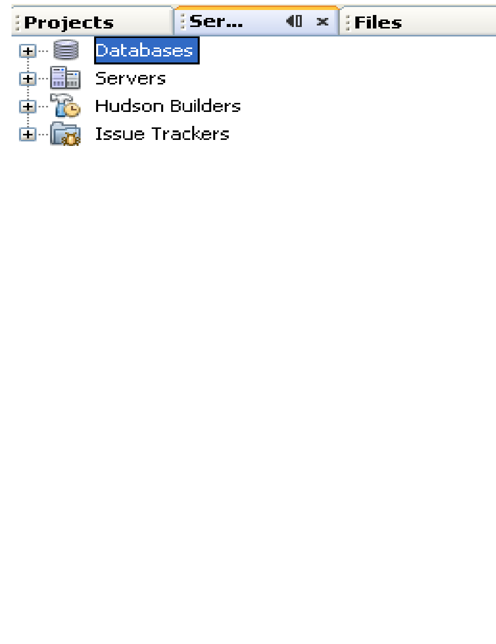
الباب الثالث قواعد البيانات

الفصل الاول : قواعد بيانات Java DB

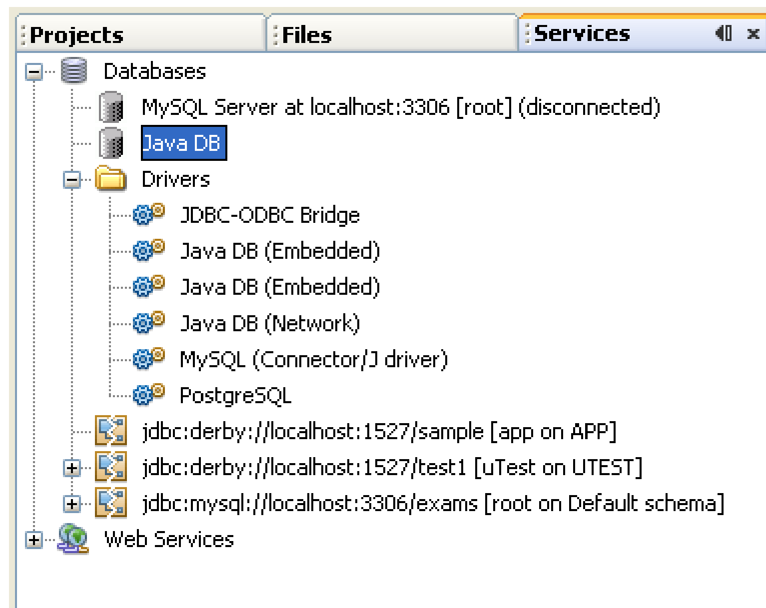
فى هذا الدرس ان شاء الله تعالى سنتعلم سويا أولى حلقات التعامل مع قواعد البيانات فى الجافا ...و درس اليوم سيكون عن كيفية انشاء قاعدة بيانات Java DB من خلال بنية **NetBeans** وهى قاعدة بيانات علائقية مفتوحة المصدر تم انشائها و ادماجها فى الجافا و التى تعتمد على نسخة Apache Derby ليتم تشغليها على سرفر تخيلى virtual serve و الذى يمكنك تشغيلة و ايقافة من خلال....NetBeans وكما سنوضح كيفية التعامل معها من إضافة و حذف و تعديل وخلافة ....وكل ذلك عن طريق المزود JDBC و الذى يتم استخدامة فى الاتصال بالكثير من قواعد البيانات مثل (أوراكل و ماى سكو...الخ)...

أولا : إنشاء قاعدة بيانات Java DB من خلال NetBeans

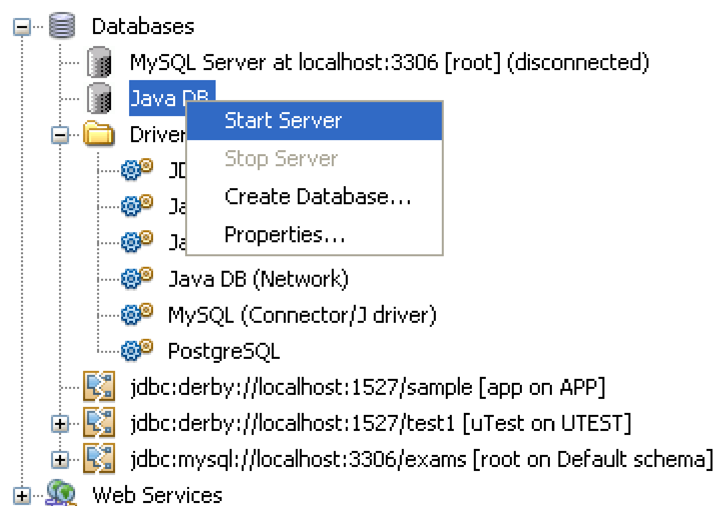
1_قم بفتح NetBeans ومن ثم قم بالضغط على التويب Services فى القوائم الجانيبة كما فى الشكل التالى:



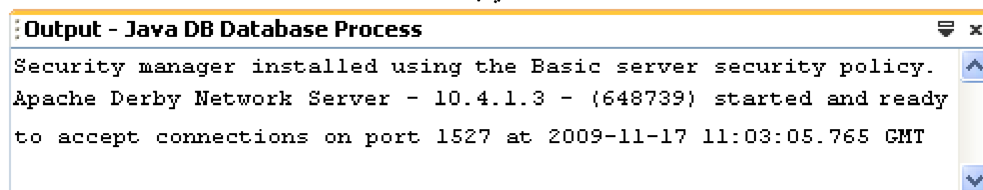
2_قم بالضغط دبل كليك على ايقونة Databases أو بالضغط على علامة ( ) التى بجوار الايقونة ليتم فتحها بالشكل التالى:



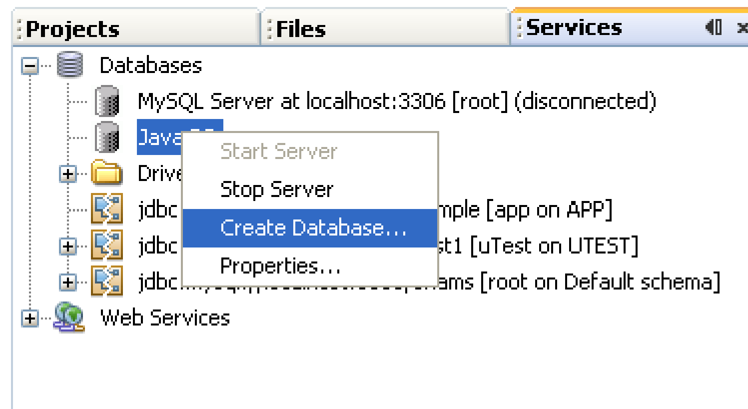
3_قم بتشغيل الخادم Virtual Server و ذلك بتحديد Java DB ثم كليك يمين ثم أختر الامر Start Server كما فى الشكل التالى:



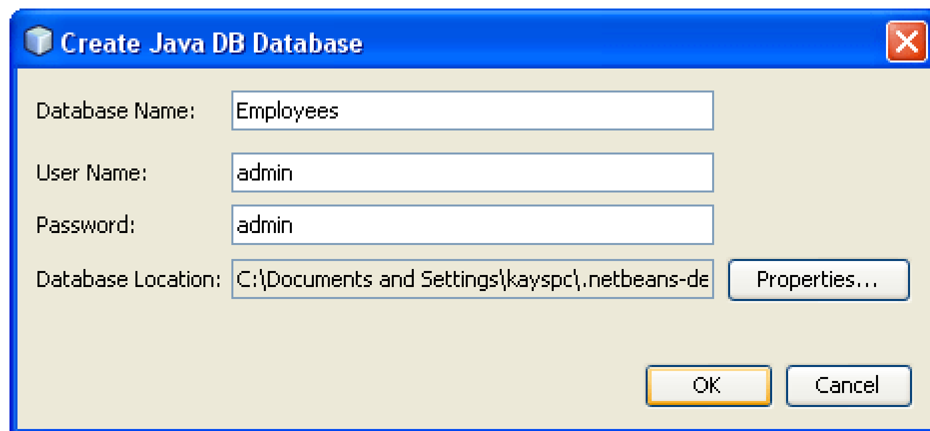
الان انظر الى نافذة الاخراج فى NetBeans و ستجد بعض الرسائل التى تخبرك بانه تم تشغيل الخادم كما فى الصورة التالية:



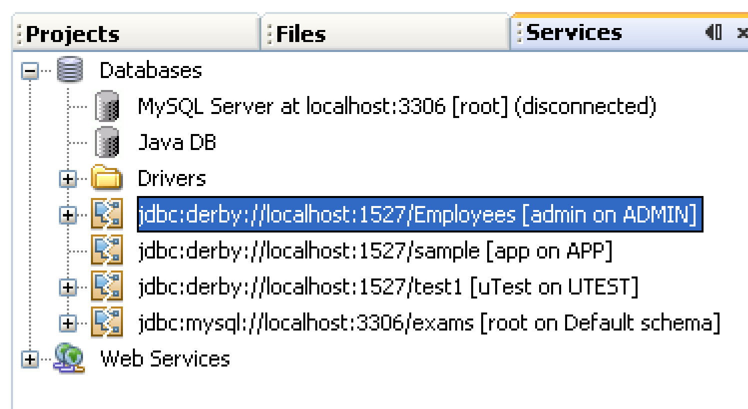
4_إنشاء قاعدة بيانات Java DB وذلك من خلال تحديد ايقونة Java DB مرة أخرى ثم كليك يمين ثم أختر الامر Create Databaseكما فى الصورة التالية:



5_عند النقر على الامر Create Database كما هو ظاهر إعلاه سوف يظهر لنا معالج إنشاء قاعدة البيانات و التى يطلب منك بعض البيانات مثل أسم قاعدة البيانات ، أسم المستخدم ، وكلمة المرور ...فعلى سبيل المثال سوف نقوم بإنشاء قاعدة بيانات تسمى "Employees" كما فى الصورة التالية:



بعد ما قمنا بإضافة كافة البيانات المطلوبة لقاعدة بيانات Employees نقوم بالضغط على زر (ok) لتظهر قاعدة البيانات فى التويب Server كما فى الشكل التالى:

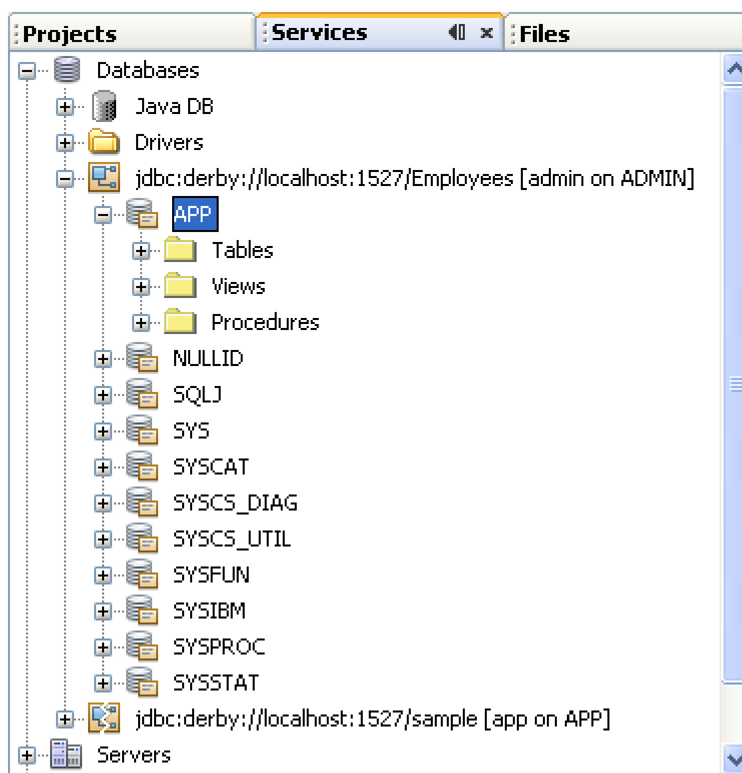


ثانيا : إنشاء جدول داخل قاعدة البيانات:

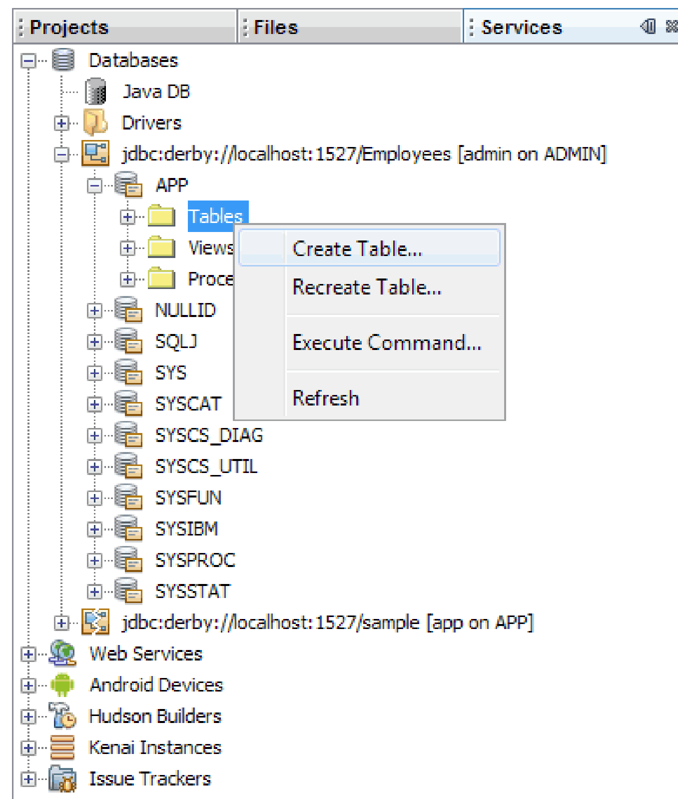
الآن بعد أن تم إنشاء قاعدة البيانات Employees ، تحتاج إلى إنشاء جدول في قاعدة البيانات. للقيام بذلك أتبع الخطوات التالية:

1_انقر بزر الماوس الأيمن على قاعدة البيانات الخاصة بك. من القائمة التي تظهر حدد الامر Connect كما فى الشكل التالى:

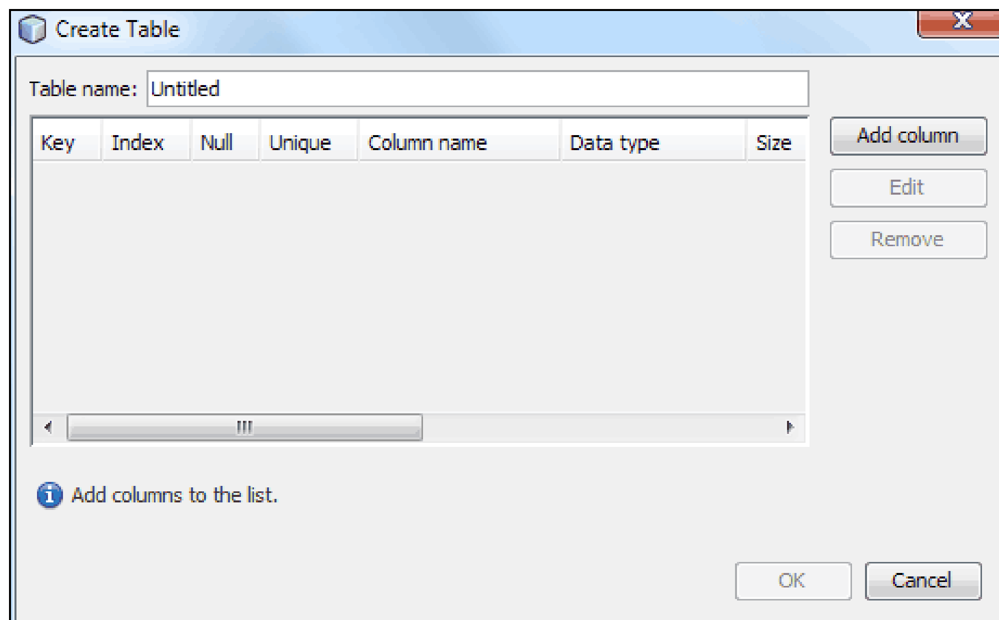
بعد ما يتم عملية الاتصال بالخادم ستجد انه قد تم انشاء ثلاث مجلدات وهى Tables , Views, Proceduresو حتى تتمكن من رؤية تلك المجلدات فى التويب Server قم بالنقر دبل كليك على قاعدة البيانات التى انشأناها Employeesلتفتح لك بعض المجلدات ثم قم بالنقر دبل كليك على المجلد APP لترى المجلدات الثلاث كما فى الشكل التالى:



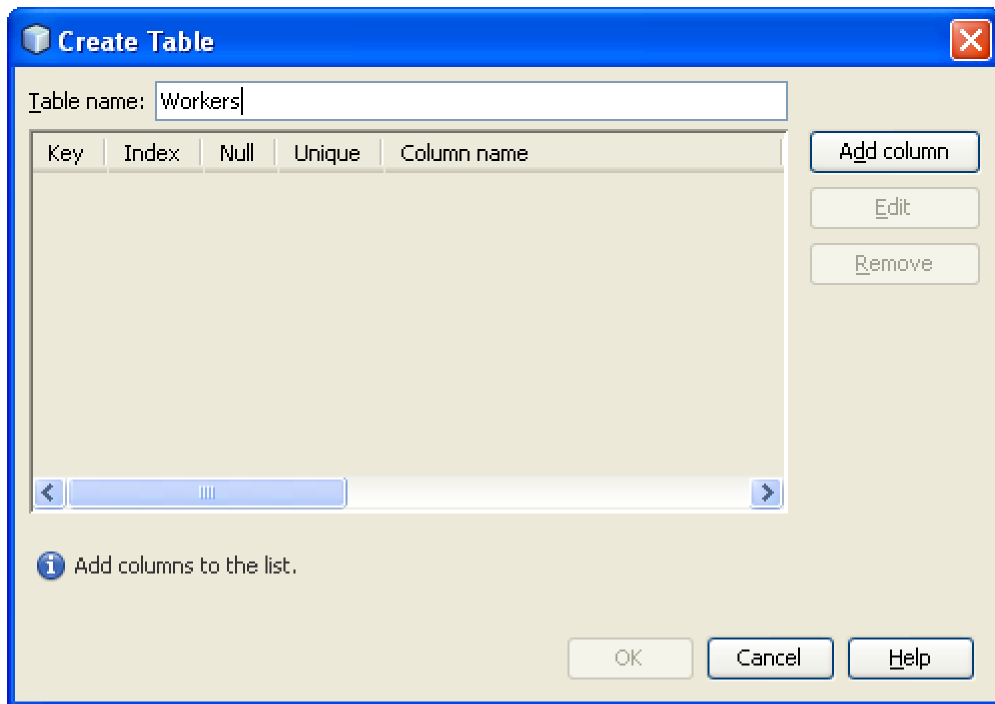
2_قم بالنقر شمال على المجلد Tables و من القائمة التي تظهر، حدد الامر Create Table كما فى الصورة التالية :



3_عند النقر كما تم شرحه أعلاه يظهر لك المربع الحوارى التالى:

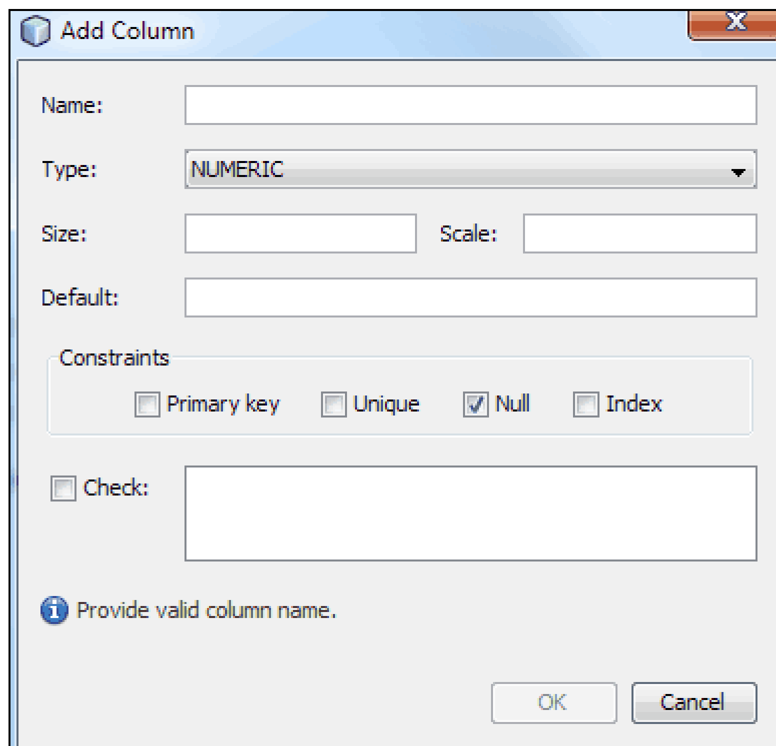


في مربع النص Table Name في الأعلى، أحذف الاسم الافتراضي..Untitled ثم اكتب اسما جديدا للجدول و ليكن مثلا....Workers و هكذا سيكون لديك جدول يسمى Workers والذي هو في قاعدة البيانات الموظفين كما فى الصورة التالية.....



ولكن لا يمكنك انقر فوق موافق حتى الآن لانشاء هذا الجدول الابعد إضافة حقول لهذا الجدول..

4_إضافة حقول للجدول و يتم ذلك عن طريق النقر فوق زر الامر Add Column فى المربع الحوارى أعلاه ليظهر لك المربع الحوارى التالى:



من المربع الحوارى السابق تستطيع إنشاء الحقول أو الاعمدة التى تريدها فى جدول البيانات ...وفى مثالنا هذا نريد إنشاء عدد ٤ حقول او اعمدة لهذا الجدول وهم (رقم العامل / الاسم الاول / الاسم الثانى / الوظيفة)...لذلك سنقوم بأنشاء أول حقل وهو رقم الموظف(ID) كما فى الشكل التالى:

ومن الملاحظ فى المربع الحوارى إعلاه انه يطلب منك فى مربع النص name أن تدخل أسم مناسب للحقل أو العمود ...وفى مربع النص TYPE و التى تحدد نوع البيانات المراد تخزينها فى هذا الحقل ...وهذا اهم المتطلبات و الباقى يتم وضع قيم افتراضية ...وعند النقر على زر OK سيظهر الحقل كما فى الشكل التالى:

سنكرر العملية لباقى الحقول مع ملاحظة التنسيق التالى:

Column Name: First_Name
Data Type: VARCHAR
Size: 20

Column Name: Last_Name
Data Type: VARCHAR
Size: 20

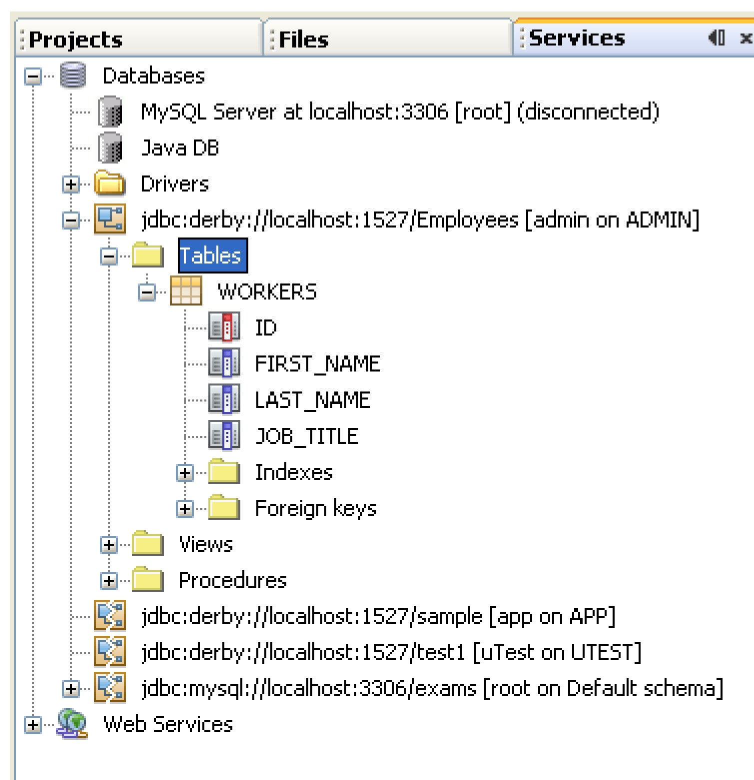
Column Name: Job_Title
Data Type: VARCHAR
Size: 40

حتى تتم كما فى الشكل التالى:

Table name: Workers   Owner: AD...

Key	Index	Null	Unique	Column name	Data type	Size	Scale
☑	☑	☐	☑	ID	INTEGER	0	0
☐	☐	☐	☐	First_Name	VARCHAR	20	0
☐	☐	☐	☐	Last_Name	VARCHAR	20	0
☐	☐	☐	☐	Job_Title	VARCHAR	40	0

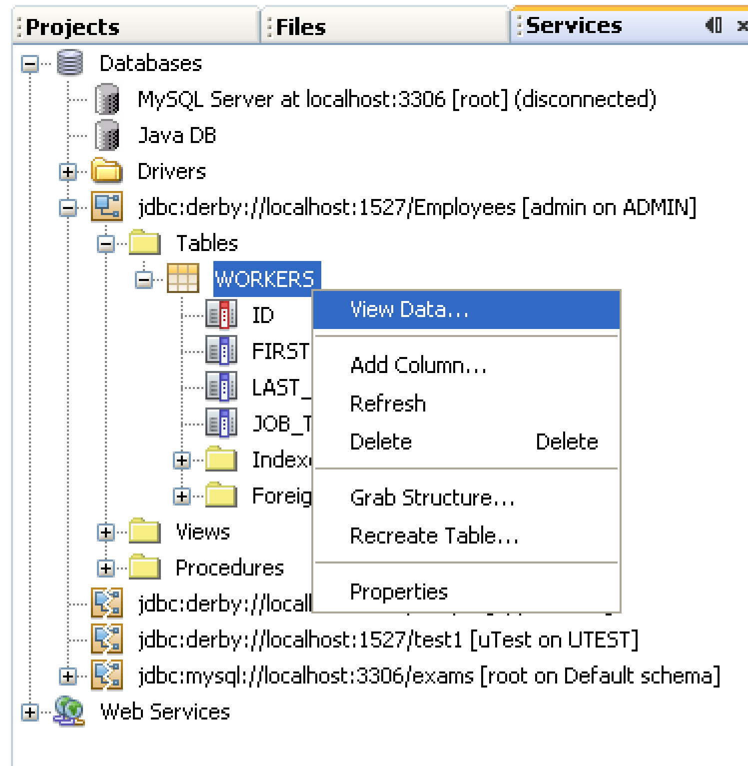
5_انقر الان فوق موافق بعد ما قد أدخلت كل المعلومات وحقول الجدول المراد إنشاؤه ليظهر كما فى الشكل التالى:



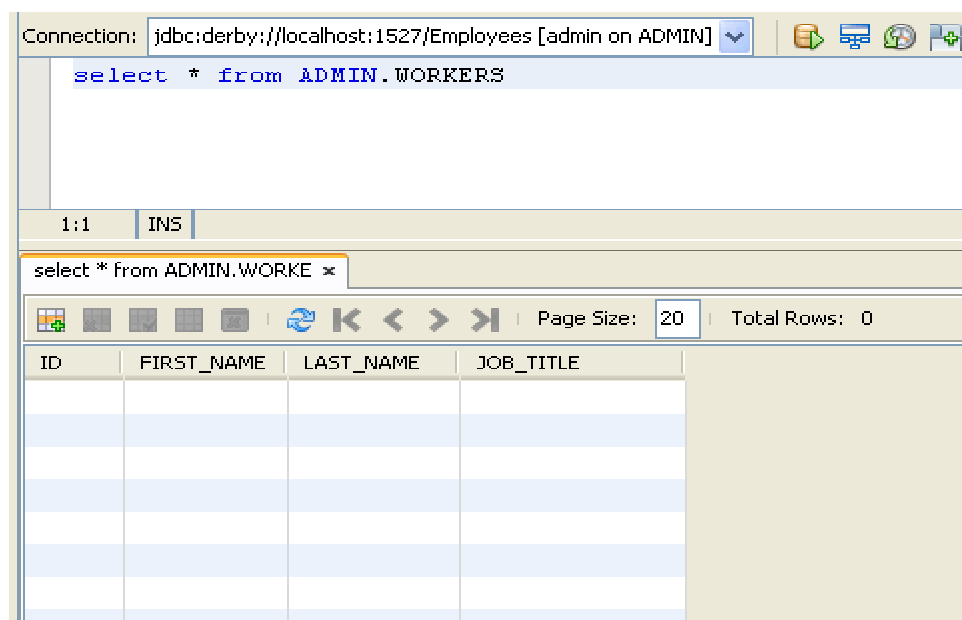
ثالثا : إدخال بيانات الى جدول قاعدة البيانات يدويا:

بعد ما تم إنشاء جدول قاعدة البيانات و حقوله تعالوا نتعلم كيفية ادخال قيم أو بيانات الى هذا الجدول يدويا كأن نفتح قاعدة البيانات و ادراج القيم بها...

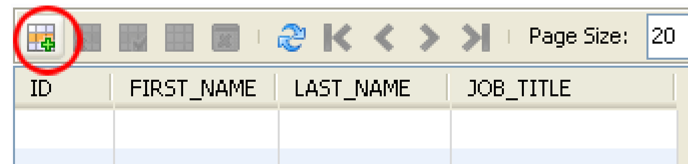
1_قم بالنقر بزر الماوس الأيمن على اسم الجدول المراد وضع قيم به و فى مثالنا هذا .Workers و من القائمة التي تظهر، حدد الامر View Data كما فى الشكل التالى:



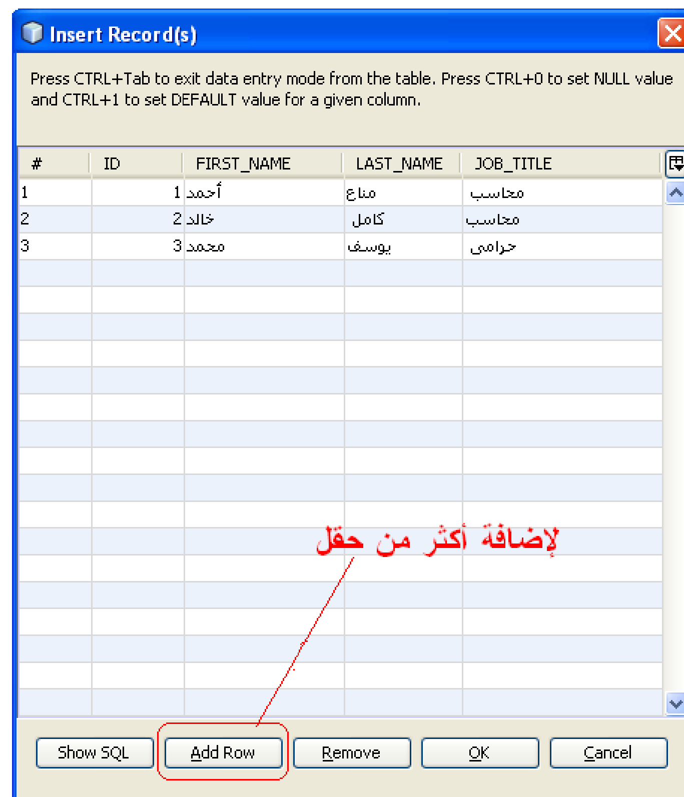
2_بعد ذلك سيظهر لك نافذة جديدة تظهر في إطار بنية NetBeans الرئيسية كما فى الشكل التالى:



و لإضافة قيم للحقول، انقر فوق رمز علامة ( )الجمع الخضراء في النصف السفلي من النافذة كما فى الشكل التالى



ليظهر لك المربع الحوارى التالى و الذى يمكنك إضافة البيانات التى ترغب فيها كما فى الشكل التالى:



رابعا : الاتصال بقاعدة البيانات

بعد ما تعلنا سويا كيفية إنشاء قاعدة بيانات Java DB و التى قد أعطينا لها أسم **Employees** و انشأنا بها جدول **Workers**و أدخلنا به بعض البيانات يدويا ....فى تلك الجزئية سنتعلم ايضا كيفية الاتصال بقاعدة البيانات تلك و جداولها و التعامل عليها بالاضافة و الحذف و التعديل ....الخ...

1_ قم بإنشاء مشروع جديد و ليكن database_console والدالة الرئسية مثلا :DBConnect كما فى الشكل التالى:

Name and Location

Project Name: database_console

Project Location: C:\Documents and Settings\kayspc\My Do | Browse...

Project Folder: Settings\kayspc\My Documents\NetBeans

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: | Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class database_console.DBConnect

```
package database_console;

public class DBConnect {

    public static void main(String[] args) {

    }
}
```

2_للاتصال قاعدة بيانات نحتاج إلى كائنات الاتصال وهى sql.Connection ,sql.DriverManager,sql.SQLException ويتم أستيراد تلك الكائنات عن طريق الكود التالى:

الرمز:

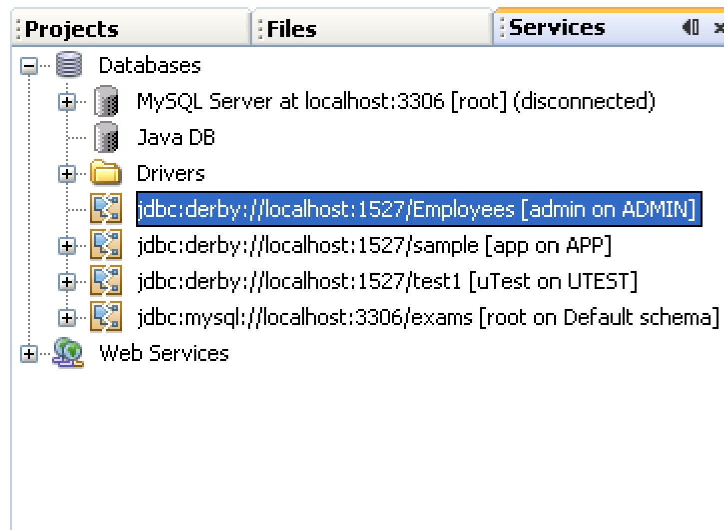
```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
```

3_نص الاتصال بقاعدة البيانات يكون عن طريق الكود التالى:

الرمز:

```
Connection con = DriverManager.getConnection( host, username, password );
```

وكما تلاحظون فى الكود أعلاه أننا انشأنا اتصال يسمى con و ذلك عن طريقة DriverManager التى به طريقة تسمى .getConnection وهو يحتاج الى اسم المضيف (الذي هو موقع قاعدة البيانات الخاصة بنا)، واسم المستخدم، وكلمة مرور.حتى يتم الاتصال بنجاح ...ويمكنك الحصول عنوان موقع قاعدة البيانات كما فى الصورة التالية:



وكما تلاحظون من الصورة فإن عنوان قاعدة البيانات المبينة أعلاه هو:

jdbc:derby://localhost:1527/Employees

لذلك سنقوم بالاعلان عن متغير نصى ليتم تخزين فية موقع قاعدة البيانات تلك و ذلك بأستخدام الكود التالى:

الرمز:

String host = "jdbc:derby://localhost:1527/Employees";

كما اننا سنقوم بتعريف متغريين أخريين الاول ليتم تخزين فية أسم المستخدم و الثانى ليتم تخزين فية رقم المرور و ذلك على النحو التالى:

String uName = "admin";
String uPass= " admin ";

وبالتالى فإن الكود الكامل سيكون على النحو التالى

```
package database_console;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnect {

  public static void main(String[] args) {

    String host = "jdbc:derby://localhost:1527/Employees";
    String uName = "admin";
    String uPass = "admin";
    Connection con = DriverManager.getConnection(host, uName, uPass);

  }
}
:
```

كما ترون في الصورة أعلاه،ستجدون خط أحمر مائل فى الكود يدل على وجود خطأ فيه ...و السبب فى ذلك هو خطأ SQLExceptionو الذى ربما يحصل نتيجة عدم وجود المضيف مثلا لذلك لابد من أستخدام التعليمةcatch ... try كما فى الشكل التالى:

```
public static void main(String[] args) {

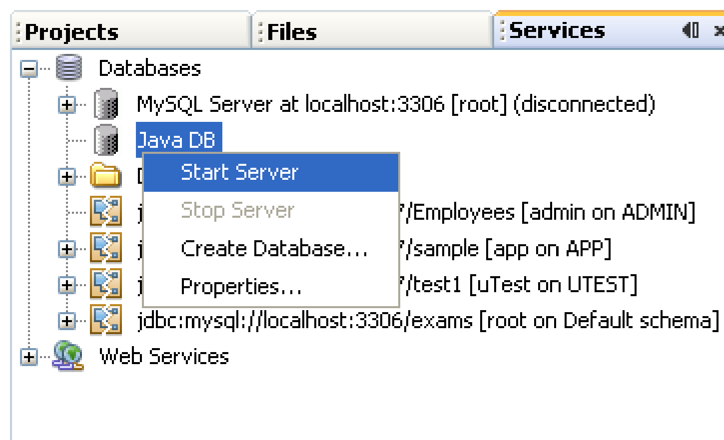
   try {
       String host = "jdbc:derby://localhost:1527/Employees";
       String uName = "admin";
       String uPass = "admin";
       Connection con = DriverManager.getConnection(host, uName, uPass);
   }
   catch ( SQLException err ) {
       System.out.println( err.getMessage( ) );
   }

}
```

الان قم بتشغيل برنامج و لاحظ ماذا سيحدث ؟.......ربما تجد رسائل الاخطاء التالية:

"java.net.ConnectException : Error connecting to server localhost on port 1527 with message Connection refused: connect."

و يحدث هذا الخطأ نتيجة أن السرفر متوقف و الحل هو تشغيله كما فى الصورة التالية :

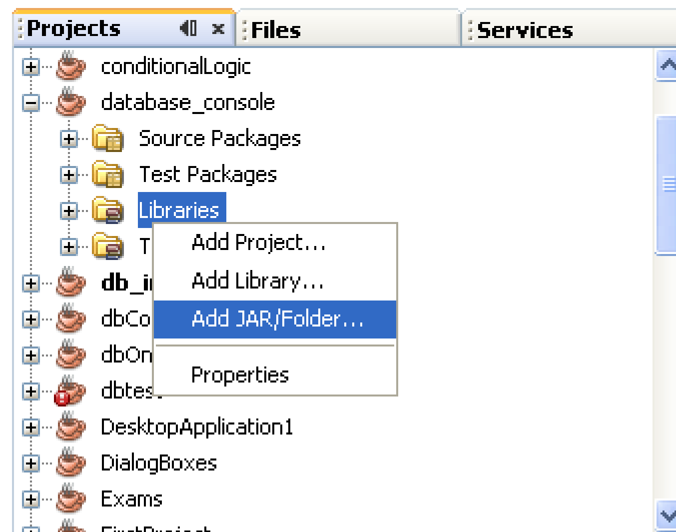


الخطأ الثانى الذى سيحدث هو ما يلى:

"No suitable driver found for jdbc:derby://localhost:1527/Employees"

و هذا الخطأ ناتج بسبب أن DriverManager يحتاج الى ملف يسمى derbyclient لنجاح الاتصال بقاعدة البيانات .....ولكى تتمكن من إضافة هذا الملف لمشروعك قم بعمل الاتى:

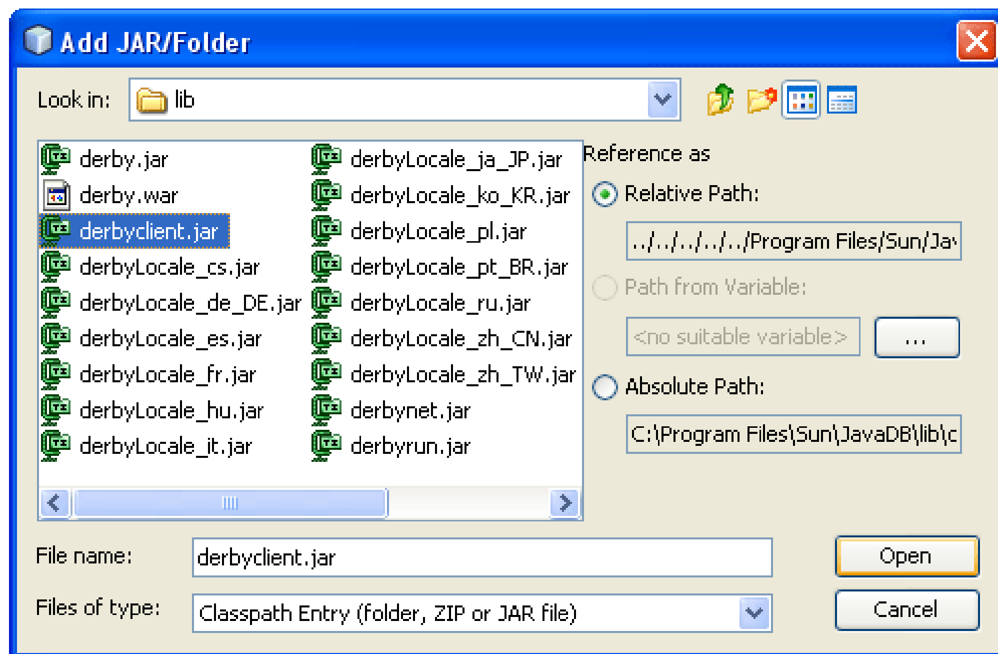
1_قم بالنقر يمين على المجلد Libraries ومن القائمة التى ستظهر أختر الامر **Add Jar/Folder** كما فى الصورة التالية:



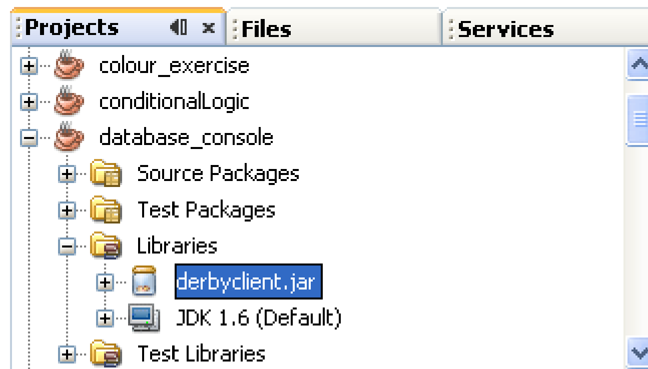
2_ سيظهر لك مربع حوارى لاضافة ملف من نوع jar قم بالبحث عن الملف... derbyclient.jar بالنسبة لى كان على المسار التالى:

C:\Program Files\Java\jdk1.7.0_01\db\lib

وقمت بإضافة للمشروع كما فى الصورة التالية:



انقر فوق فتح، وسيتم إضافة الملف إلى مكتبة المشروع الخاص بك كما فى الشكل التالى:



الان يمكنك إعادة تشغيل البرنامج ولن تجد أى رسائل أخطاء اخرى معنى ذلك أنه قد تم الاتصال بقاعدة البيانات بنجاح

خامسا : قراءة محتويات جدول البيانات

بعد ما قمنا بالاتصال بقاعدة البيانات، فإن الخطوة التالية هي للوصول إلى الجدول في قاعدة البيانات وهو فى مثالنا هذا.Workers ...و لكى يتم الاتصال بهذا الجدول نحتاج إلى تنفيذ عبارةSQL ، ومن ثم معالجة كافة الصفوف والأعمدة التي تم إرجاعها....وذلك على النحو التالى:

1_قم باستيراد الكائن sql.Statement والكائن sql.ResultSet و الكائن sql.ResultSet و ذلك عن طريق الكود التالى:

```
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;
```

2_فى منطقة معالجة الاستثناءات try..... catch قم بوضع الكود التالى:

```
Statement stmt = con.createStatement( );
```

هنا قمنا بإنشاء الكائن Statement بأسم stmt

3_تحت الكود السابق قم بانشاء جملة استعلام للمرور بجدول البيانات Workers و ذلك عن طريق الكود التالى:

```
String SQL = "select * from APP.WORKERS";
```

وجملة الاستعلام السابقة تقوم بتحديد كافة السجلات فى الجدول Workers

4_نقوم بإنشاء مؤشر للمرور على قاعدة البيانات و معالجة بياناتها وليكن rs ثم تمير جملة الاستعلام السابقة و ذلك عن طريق الكود التالى:

```
ResultSet rs = stmt.executeQuery( SQL );
```

وكما تلاحظون فى الكود أعلاه أننا قد انشأنا المؤشر rs و هو الوسيلة لتخزين ومعالجة السجلات التي يتم إرجاعها من استعلام SQLو يأخذ ثلاث انواع هما:

TYPE_FORWARD_ONLY_و هذا النوع يقوم بالمرور على السجلات للامام فقط من البداية الى النهاية.
TYPE_SCROLL_SENSITIVE_و هذا النوع يقوم بالمرور على السجلات للإمام و للخلف مع الكشف عن أي تغييرات يتم إجراؤها على السجلات.

TYPE_SCROLL_INSENSITIVE_وهذا النوع يقوم بالمرور على السجلات للإمام و للخلف مع عدم الكشف عن أي تغييرات يتم إجراؤها على السجلات.

ولكى تستطيع ان تحدد نوع المؤشر الخاص بلك اذهب الى الاقواس المستديرة فى جملة التالية:

```
Statement stmt = con.createStatement( );
```

و بداخل تلك الاقواس المستديرة يمكنك وضع نوع المؤشر على النحو التالى:

```
Statement stmt = con.createStatement( RecordSet.TYPE_SCROLL_SENSITIVE );
```

وذلك لو تركنا تلك الاقواس فارغة بدون تحديد المؤشر فإن البرنامج سوف يعتمد على المؤشر الافتراضى للسجلات وهو فى هذه الحالةTYPE_FORWARD_ONLY ....ومع ذلك لم ننتهى بعد من الكود السابق و ذلك لكوننا فى حالة استخدامنا النوعين الاخريين من المؤشرات فيجب تحديد ما إذا كان هذا المؤشر للقراءة فقط ام قابل للتحديث و ذلك باستخدام الثوابت التالية

```
ResultSet. CONCUR_READ_ONLY
ResultSet. CONCUR_UPDATABLE
```

و بالتالى فإن الكود النهائى سيصبح كما يلى:

```
Statement stmt = con.createStatement( RecordSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
```

لكن فى مثالنا هذا لن نقوم الان بالاضافة و التعديل و خلافة ..و بالتالى سنكتفى بالمؤشر الافتراضىTYPE_FORWARD_ONLY

4_إسناد حقول جدول لمتغيرات Workers حتى يستنى لنا طباعتها و ذلك عن طريقة حلقة تكرارية كما فى الكود التالى :

```
while ( rs.next( ) ) {
int id_col = rs.getInt("ID");
String first_name = rs.getString("First_Name");
String last_name = rs.getString("Last_Name");
String job = rs.getString("Job_Title");
String p= id_col + " " + first_name + " " + last_name + " " + job ;
System.out.println(p);

}
```

ويمكنك ملاحظة فى الكود أعلاه أننا قد استخدمنا حلقة تكرارية للمرور على جميع سجلات الجدول ...واننا قد قمنا بالاعلان عن متغيرات لكل عمود فى الجدول على حسب نوع البيانات لكل عمود و هذا هو الكود بالكامل:

```
package database_console;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;
```

```
public class DBConnect {


   public static void main(String[] args) {
    try {
      String host = "jdbc:derby://localhost:1527/Employees";
      String uName = "admin";
      String uPass= "admin";

      Connection con = DriverManager.getConnection( host, uName, uPass);
      Statement stmt = con.createStatement( );
      String SQL = "select * from APP.WORKERS";
      ResultSet rs = stmt.executeQuery( SQL );

 while ( rs.next( ) ) {
int id_col = rs.getInt("ID");
String first_name = rs.getString("First_Name");
String last_name = rs.getString("Last_Name");
String job = rs.getString("Job_Title");
String p= id_col + " " + first_name + " " + last_name + " " + job ;
 System.out.println(p);

}
    }
    catch(SQLException e){

      System.out.println(e.getMessage());
    }    }   }
```

مثال للاتصال بقاعدة البيانات فى الواجهة الرسومية

أهلا و سهلا إخوانى الكرام فى الجزء الاخير من هذا الدرس وهو عن كيفية الاتصال بقاعدة البيانات فى الوجهة الرسومية ...تلك الجزئية سهلة بإذن الله تعالى طالما فهمت الاجزاء الاولى لهذا الدرس فهى نفس الاكواد البرمجية مع تعديل بسيط و ذلك على النحو التالى:

1_ قم إنشاء نافذة رسومية تحتوي على أزرار وحقول نص كما فى الشكل التالى:

2_قم بحذف النص الافتراضي للحقول النص وترك كل منهم فارغ...وقم ايضا بتغيير أسماء المتغيرات الافتراضية للحقول نص على ما يلي:

```
textID
textFirstName
textLastName
textJobTitle
```

وطبعا تلك الاسماء سوف يتم التعامل عليها فى صفحة الكود عند كتابته....

3_الان سنقوم باستيراد الكائنات اللازمة و التى قد تعرفنا عليها أعلاه للاتصال بقاعدة البيانات Employees و جدولها Workersوذلك عن طريق كود واحد فقط و هو التالى:

```
import java.sql.*;
```

بهذا الكود البسيط قمنا بإستيراد جميع الكائنات الموجودة بالفئة sql فبدلا من اتسيراد كل كائن على حده كما فعلنا إعلاه أختصرنا تلك الاكواد بهذا الكود البسيط فالرمز النجمة (*) تعنى جميع كائنات الفئة..

4_نقوم بإستيراد الكائن JOptionPane و ذلك لعرض رسائل الاخطاء التى قد تحدث بالشكل الرسومى المتعارف عليه بدلا من System.out.println و يتم الاستيراد باستخدام الكود التالى:

```
import javax.swing.JOptionPane;
```

5_نقوم بتعريف متغيرات الاتصال بقاعدة البيانات التالية:

```
Connection con;
Statement stmt;
ResultSet rs;
```

ويمكنك ملاحظة أننا قمنا بتعريف تلك المتغيرات على مستوى الفئة حتى نتمكن من استخدامها فى أى حدث من الاحداث داخل الفئة

6_نقوم بإنشاء طريقة و ليكن مثلا DoConnect و ذلك لوضع كافة التعليمات البرمجية الخاصة بالاتصال بقاعدة البيانات وجدولها ....و يتم إنشاء الطريقة باستخدام الكود التالى:

```
public void DoConnect( ) {


}
```

6_بداخل الطريقة سنقوم بوضع الاكواد اللازمة للاتصال بقاعدة البيانات و جدولها كما يلى:

_كود الاتصال بقاعدة البيانات:

```
String host = "jdbc:derby://localhost:1527/Employees";
    String uName = "admin";
    String uPass= "admin";

    con = DriverManager.getConnection( host, uName, uPass);
```

-كود الاتصال بجدول البيانات:

```
stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE );
      String SQL = "select * from APP.WORKERS";
      rs = stmt.executeQuery( SQL );
```

ستلاحظ أننا قد إضافنا بعد التغيرات فى الكود السابق للاتصال بجدول البيانات عن الكود إعلاه فى الجزيئة السابقة ..فقد حدد نوع المؤشر و أنه لتحديث البيانات و ذلك لإننا سوف نتعامل مع قاعدة البيانات بالاضافة و الحذف و التعديل على بياناتها

_كود إسناد حقول الجدول الى مربعات النصوص فى النافذة:

```
rs.next( );
    int id_col = rs.getInt("ID");
    String id =Integer.toString(id_col);
    String first_name = rs.getString("First_Name");
    String last_name = rs.getString("Last_Name");
    String job = rs.getString("Job_Title");

    textID.setText(id);
    textFirstName.setText(first_name);
    textLastName.setText(last_name);
    textJobTitle.setText(job);
```

و بالنظرة السريعة على الكود سنجد أننا قمنا بالاعلان عن المتغير الرقمى الصحيح id_col و ذلك لتخزين قيمة الحقل الموجود فى جدول البيانات الذى نوعه ايضا رقمى....(integer) لكن المشكلة ان حقول النص فى الجافا لا تقبل الا قيم نصية بها لذلك قمنا بالاعلان عن المتغير النصى (id) لتخزين فية قيمة المتغير id_col بعد تحويلة الى متغير نصى باستخدام الامر Integer.toString

_وضع كود معالجة الاستثناءات try.... catch و بالتالى فإن الكو الكامل سيصبح كما يلى:

```
public void DoConnect( ) {
 try {
   String host = "jdbc:derby://localhost:1527/Employees";
   String uName = "admin";
   String uPass= "admin";
   Connection con = DriverManager.getConnection( host, uName, uPass);
   Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_UPDATABLE );
   String SQL = "select * from APP.WORKERS";
   ResultSet rs = stmt.executeQuery( SQL );

  rs.next( );
  int id_col = rs.getInt("ID");
  String id =Integer.toString(id_col);
  String first_name = rs.getString("First_Name");
  String last_name = rs.getString("Last_Name");
  String job = rs.getString("Job_Title");

 textID.setText(id);
 textFirstName.setText(first_name);
 textLastName.setText(last_name);
 textJobTitle.setText(job);

  catch (SQLException err) {
  JOptionPane.showMessageDialog(main.this, err.getMessage());
 }
```

7_الان قم بوضع ;( )DoConnect فى الدالة الرئيسية للبرنامج كما فى الشكل التالى:

_8قم بتشغيل برنامج و ستجد قيم جدول البيانات قد ظهرت فى مربعات النصوص كما فى الشكل التالى:

_9برمجة ازرار التنقل بين السجلات **التالى / السابق/ الاول / الاخير**..و ذلك على النحو التالى:

<u>برمجة زر التالى</u>

المطلوب من هذا الزر هو عند النقر عليه يجب أولا التحقق ما إذا كان هناك سجل التالي للانتقال إليه أم لا وثانيا، إذا كان هناك السجل التالي، يقوم بعرضه في الحقول النصية ....ويتم ذلك باستخدام الكود التالى _ قم بالنقر على هذا الزر ليتم فتح صفحة الكود ثم اكتب الكود التالى:

```
try {

if ( rs.next( ) ) {
int id_col = rs.getInt("ID");
    String id =Integer.toString(id_col);
    String first_name = rs.getString("First_Name");
    String last_name = rs.getString("Last_Name");
    String job = rs.getString("Job_Title");

    textID.setText(id);
    textFirstName.setText(first_name);
    textLastName.setText(last_name);
    textJobTitle.setText(job);
}
else {
rs.previous( );
JOptionPane.showMessageDialog(main.this, "End of File");
}
}
catch (SQLException err) {
JOptionPane.showMessageDialog(main.this, err.getMessage());
}
```

_فى السطر الاول من الكود استخدمنا تعليمة try...cacth و ذلك لمعالجة الاخطاء التى ستظهر فى حالة عدم وجود سجل للانتقال إلية ...ثم مررنا الرسالة (End of File) اليها حتى يتم اظهارها للمستخدم لاخبارة بذلك

_فى السطر الثانى أستخدمنا الجملة الشرطية if.....else وذلك للتحق من وجود سجل للانتقال الية ....فى حالة وجودة يتم الانتقال الية و إسناد حقول الجدول الى مربعات النصوص...

<u>برمجة زر السابق</u>

الكود الخاص بهذا يشبة الى حد كبير الكود الخاص بزر التالى إعلاه مع فرق بسيط ...فبدلا من استخدام **rs.Next بعد كملة if سوف نتسخدم... rs.previous و ايضا بعد كلمة else بدلا من استخدام rs.previous سوف نتسخدم rs.nextو بالتالى فإن الكود لهذا الزر كما يلى:**

```
try {

if ( rs.previous() ) {
int id_col = rs.getInt("ID");
    String id =Integer.toString(id_col);
    String first_name = rs.getString("First_Name");
    String last_name = rs.getString("Last_Name");
    String job = rs.getString("Job_Title");

    textID.setText(id);
    textFirstName.setText(first_name);
    textLastName.setText(last_name);
    textJobTitle.setText(job);
}
else {
rs.next();
JOptionPane.showMessageDialog(main.this, "End of File");
}
}
catch (SQLException err) {
JOptionPane.showMessageDialog(main.this, err.getMessage());
}
```

<u>برمجة زر الامر الاول</u>

المطلوب من هذا الزر هو عند النقر علية الانتقال الى اول حقل فى جدول قاعدة البيانات ...و بالتالى فالكود سهل جدا فقط استخدم الامر rs.first ثم إسناد الحقول الى مربعات النصوص ....و بالتالى فالكود سيكون على النحو التالى:

```
try {

 rs.first() ;
int id_col = rs.getInt("ID");
    String id =Integer.toString(id_col);
    String first_name = rs.getString("First_Name");
    String last_name = rs.getString("Last_Name");
    String job = rs.getString("Job_Title");

    textID.setText(id);
```

```
    textFirstName.setText(first_name);
    textLastName.setText(last_name);
    textJobTitle.setText(job);
}
catch (SQLException err) {
JOptionPane.showMessageDialog(main.this, err.getMessage());
}
```

برمجة زر (الاخير) المطلوب من هذا الزر هو الانتقال الى اخر سجل فى جدول قاعدة البيانات ....و هو نفس الكود أعلاه مع تغير فقط rs.first الى rs.last

10_ برمجة ازرار التعامل مع البيانات

برمجة زر جديد

المطلوب من هذا الزر عند النقر عليه مسح القيم الموجودة فى مربعات النصوص و ذلك لتهيئة ادخال القيم الجديدة و ذلك عن طريق الكود التالى:

```
try {

    textFirstName.setText("");
    textLastName.setText("");
    textJobTitle.setText("");
    textID.setText("");

}
catch (SQLException err) {
    System.out.println(err.getMessage());
}
```

برمجدة زر الامر حفظ

المطلوب من هذا الزر عند النقر عليه فتح سجل فى جدول قاعدة البيانات وحفظ القيم الموجودة فى مربعات النصوص فى هذا السجل ...ولكى تتمكن من ذلك يجب تحريك المؤشر إلى ما يسمى **Insert Row** و هذا يخلق سجل فارغ في .resultsetثم يتم إسناد القيم فى مربعات النصوص إلى ResultSe و ذلك على النحو التالى:

بداية نقوم بالاعلان عن المتغيرات و إسناد قيم مربعات النصوص اليها على حسب نوع كل حقل و ذلك على النحو التالى :

```
String id1 =textID.getText();
 int newid=Integer.parseInt(id1);

 String first1 =textFirstName.getText();
 String last1 =textLastName.getText() ;
 String job1 = textJobTitle.getText();
```

ثانيا :بعد ذلك نقوم باسناد القيم المخزنة فى المتغيرات أعلاه الى حقول الجدول المراد إضافة البيانات الية ثم نقوم بعملية إدراج لتلك الحقول و ذلك عن طريق الكود التالى:

```
try{
   rs.moveToInsertRow( );

   rs.updateInt("ID", newid);
   rs.updateString("First_Name", first1);
   rs.updateString("Last_Name", last1);
   rs.updateString("Job_Title", job1);

   rs.insertRow( );
```

ثالثا نقوم بإغلاق الكائن stmt و الكائن rs و إظهار رسالة للمستخدم تخبرة باتمام عملية الحفظ و الاضافة و ذلك عن طريق الكود التالى:

```
 stmt.close();
   rs.close();
JOptionPane.showMessageDialog(null, "");
```

و الكود النهائى لزر الامر حفظ سيكون على النحو التالى:

```
public void DoConnect( ) {
  try {
    String host = "jdbc:derby://localhost:1527/Employees";
    String uName = "admin";
    String uPass= "admin";
    con = DriverManager.getConnection( host, uName, uPass);
     stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_UPDATABLE );
    String SQL = "select * from APP.WORKERS";
     rs = stmt.executeQuery( SQL );

   rs.next( );
   int id_col = rs.getInt("ID");
   String id =Integer.toString(id_col);
   String first_name = rs.getString("First_Name");
   String last_name = rs.getString("Last_Name");
   String job = rs.getString("Job_Title");

  textID.setText(id);
  textFirstName.setText(first_name);
  textLastName.setText(last_name);
  textJobTitle.setText(job);
}
   catch (SQLException err) {
   JOptionPane.showMessageDialog(main.this, err.getMessage());
   }
}
```

_**برمجة زر الامر حذف** عن طريق الكود التالى:

```
 try{
rs.deleteRow();

stmt.close();
rs.close();
}
catch (SQLException err) {
    JOptionPane.showMessageDialog(main.this, err.getMessage());
    }
```

الفصل الثانى : قواعد بيانات MySQL

فى الفصل السابق تعلمنا كيفية إنشاء قاعدة بيانات Java DB من خلال NetBeans و كيفية التعامل معها أما فى هذا إن شاء الله تعالى سنتعلم كيفية إنشاء قاعدة بيانات MySQL و الاتصال بها سواء كانت تلك القاعدة على القرص الصلب أو على شبكة الانترنت...

أولا التعريف بقواعد بيانات: MySQL

قاعدة بيانات MySQL هى عبارة عن قاعدة بيانات علاقية من اشهر قواعد البيانات مفتوحة المصدر التى يمكنك أستخدامها مجانا و جميع تطبيقات الويب تستخدم تلك القاعدة لتحزين البيانات و يمكنك استخدامها ايضا فى تطبيقات سطح المكتب..

ثانيا إنشاء قاعدة بيانات MySQL من خلال NetBeans على السرفر المحلى (القرص الصلب):

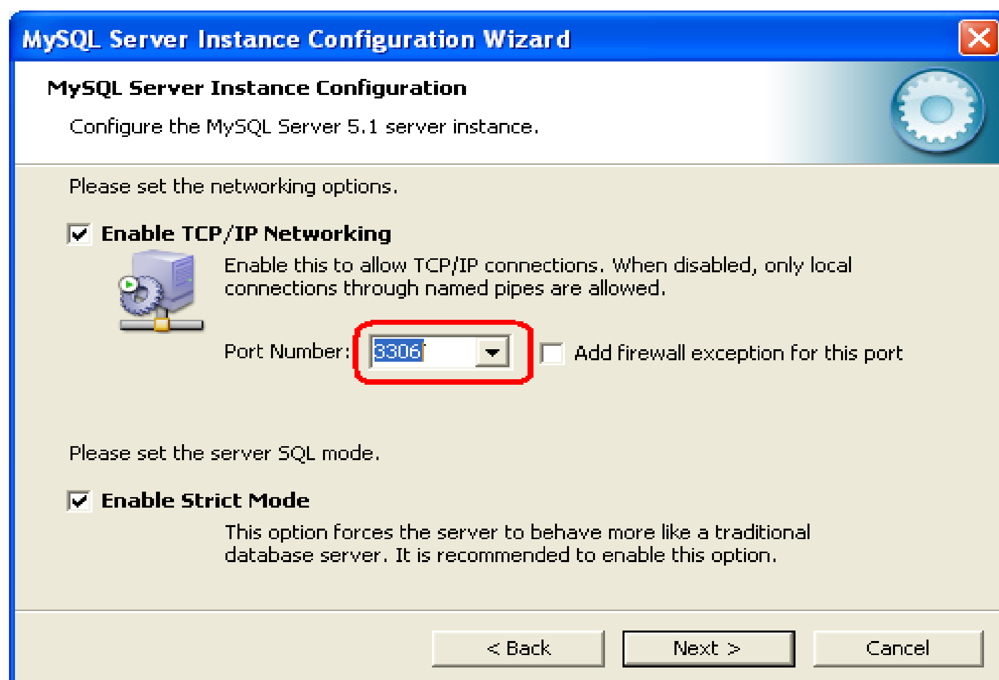
توجد طرق عديدة لانشاء قاعدة بيانات MySQL منها أستخدام موجهة الاوامر Dos و ايضا بعض البرنامج الاخرى التى تقوم بذلك .....لكن دراستنا ستكون عن كيفية إنشائها من خلال بنية NetBeans وذلك على النحو التالى:

1_سنحتاج الى تثبيت حزمة mysql-server و حزمة mysql-client و تلك الحزمتين ستجدهما فى الملف mysql-essential-5.1.5 و يمكنك تحميله من الرابط التالى:

http://downloads.skysql.com/archive/signature/p/mysql/f/mysql-essential-5.1.52-win32.msi/v/5.1.52

http://dev.mysql.com/downloads/mysql/5.1.html

2_بعد الانتهاء من تحميل الملف السابق قم بتثبته و يكفيك لذلك الضغط على زر..(next) و لكن كل ما اريده منك هو عند ظهور الشاشة التالية:

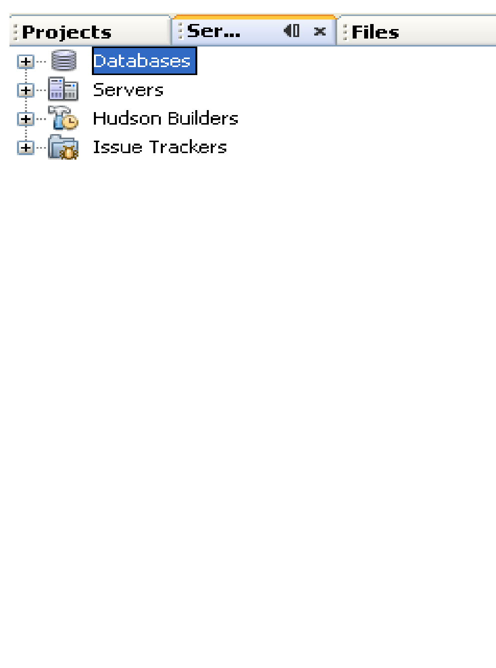


اريد منك أن تعلم و تحفظ رقم البور أو المنفذ الخاص بك وهو نفسه الموجود الصورة أعلاه.....و ايضا عند ظهور الشاشة التى تطالبك بوضع كلمة المرور للسرفر أن حفظها كما فى الشكل التالى:

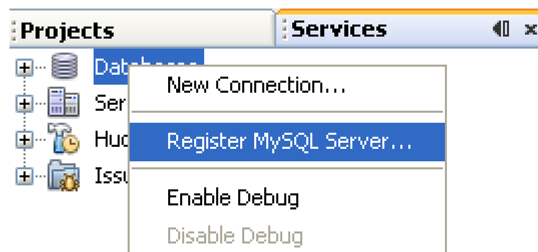


رقم المرور الذى سوف تقوم بادخاله ستخدمة عند انشاء قاعدة البيانات بواسطة NetBeans

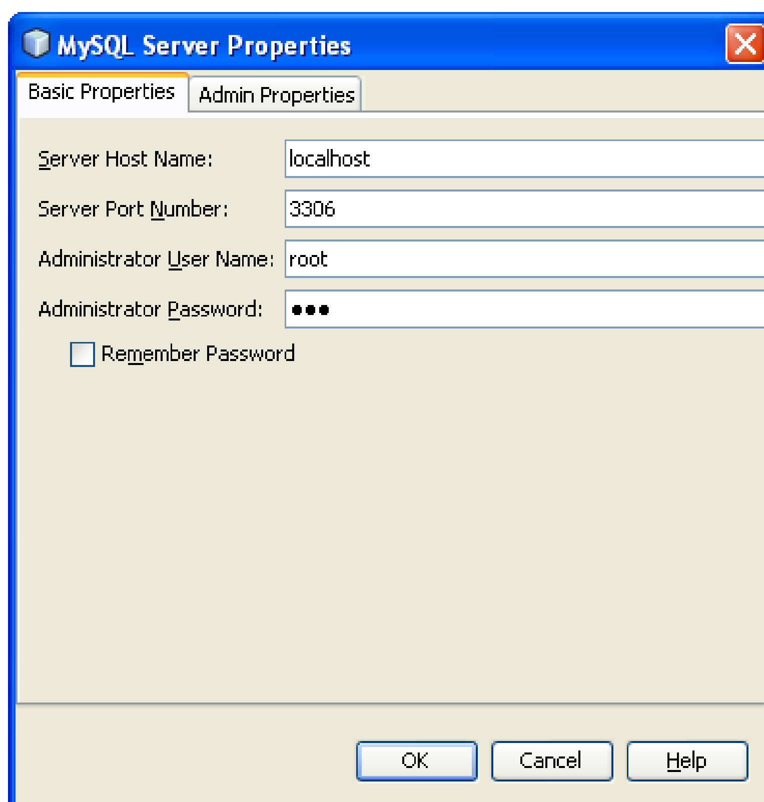
3_قم بفتح NetBeans ومن ثم قم بالضغط على التويب Services فى القوائم الجانيبة كما فى الشكل التالى :



4_قم بتحديد ايقونة Databases ثم كليك يمين ومن القائمة التى سوف تظهر أختر الامر register mysql serviceكما فى الشكل التالى:

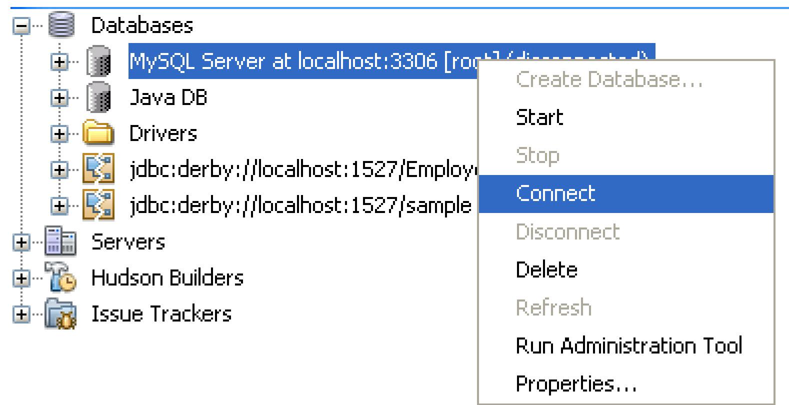


5_عند النقر على الامر register mysql service كما هو ظاهر أعلاه سيظهر لك المربع الحوارى التالى:

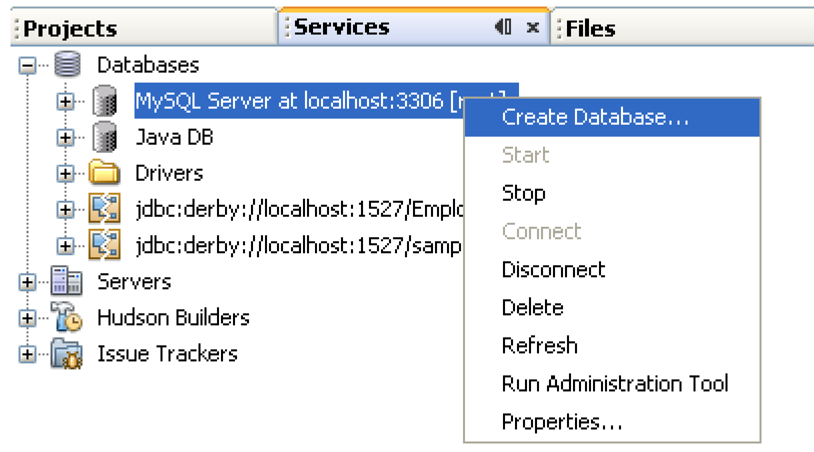


يطالبك هذا المربع service hots name و هو فى مثالنا هذا... localhost وايضا رقم المنفذ service port وطبعا هو نفس رقم المنفذ الذى حددناه اثناء تنصيب...service اما اسم المستخدم فهو root وكلمة المرور هى نفس كلمة المرور التى حددناها اثناء تنصيب السرفر.... بعد اتمام ادخال البيانات انقر على زر الامر (موافق)

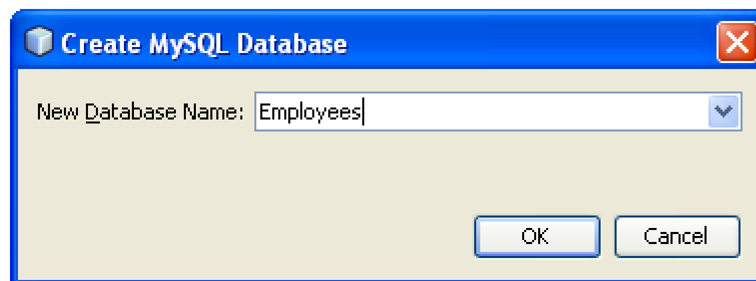
6_ بعد النقر على(ok) ستجد فى التبويب Services ايقونة mysql Services at:
.... localhost:3306[rootقم بتحديد تلك الايقونة ثم كليك يمين و من القائمة التى ستظهر أختر الامر connet كما فى الشكل التالى:



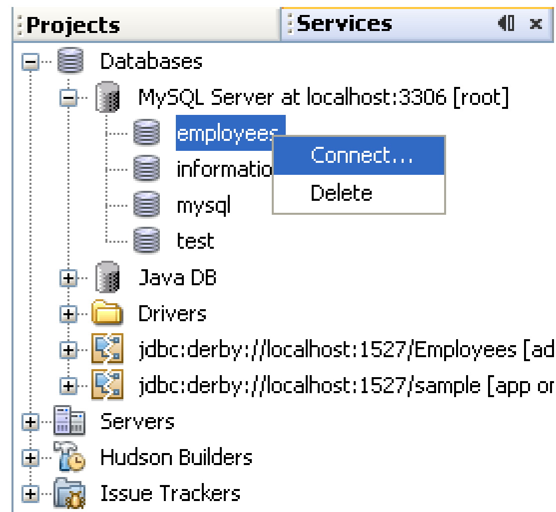
7_بعد تشغيل السرفر عاود الكرة مرة أخرى بتحديد الايقونة ثم كليك يمين و من القائمة التى ستظهر اختر الامر Create Database كما فى الصورة التالية:



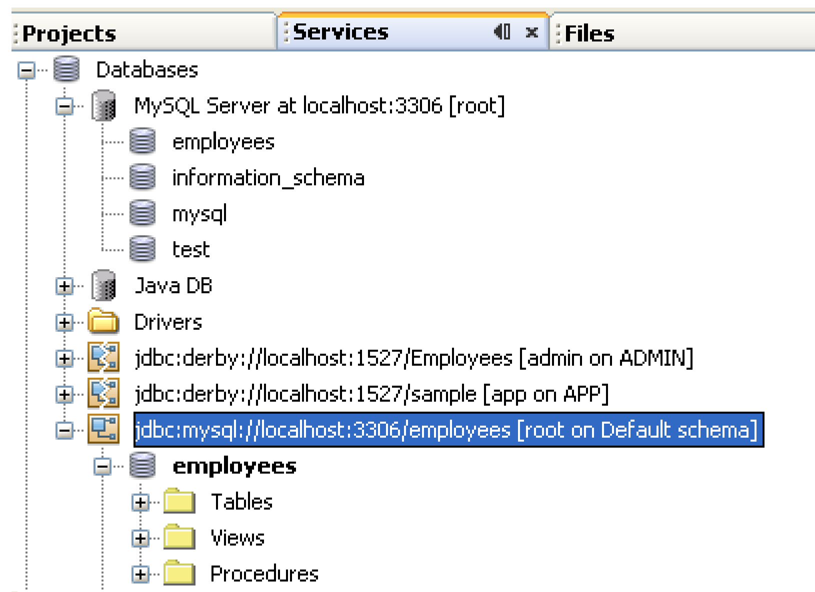
8_سيظهر لك المربع الحوارى التالى يطالبك بإدخال اسم لقاعدة البيانات المراد انشائها و ليكن مثلاً Employees كما فى الصورة التالية:



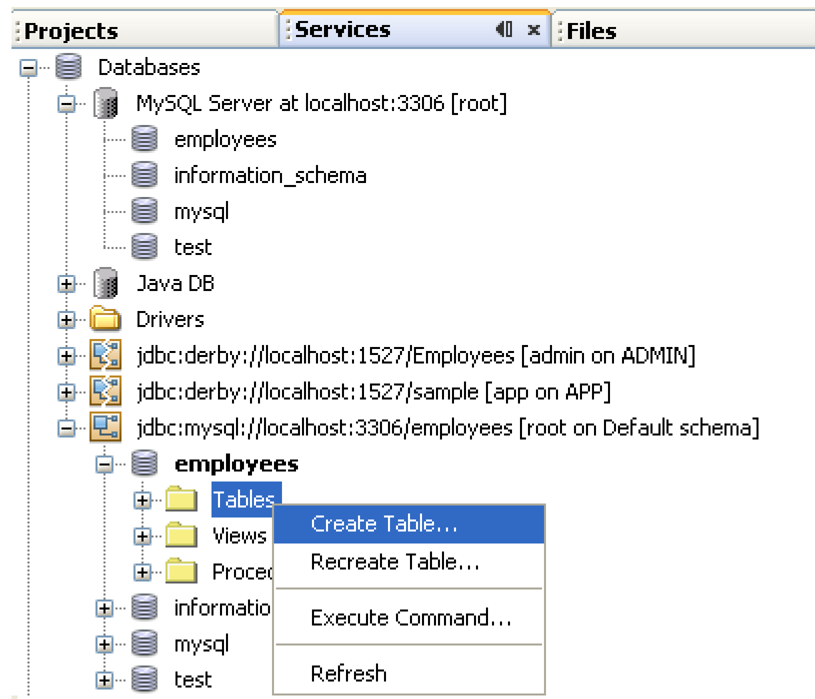
9_قم الان بالنقر دبل كليك على ايقونة mysql Services at localhost:3306[root ستجد تحتها قاعدة البيانات التى قمنا بإنشاها ...قم بتحديها ثم كليك يمين و من القائمة التى ستظهر أختر الامر connet كما فى الشكل التالى:



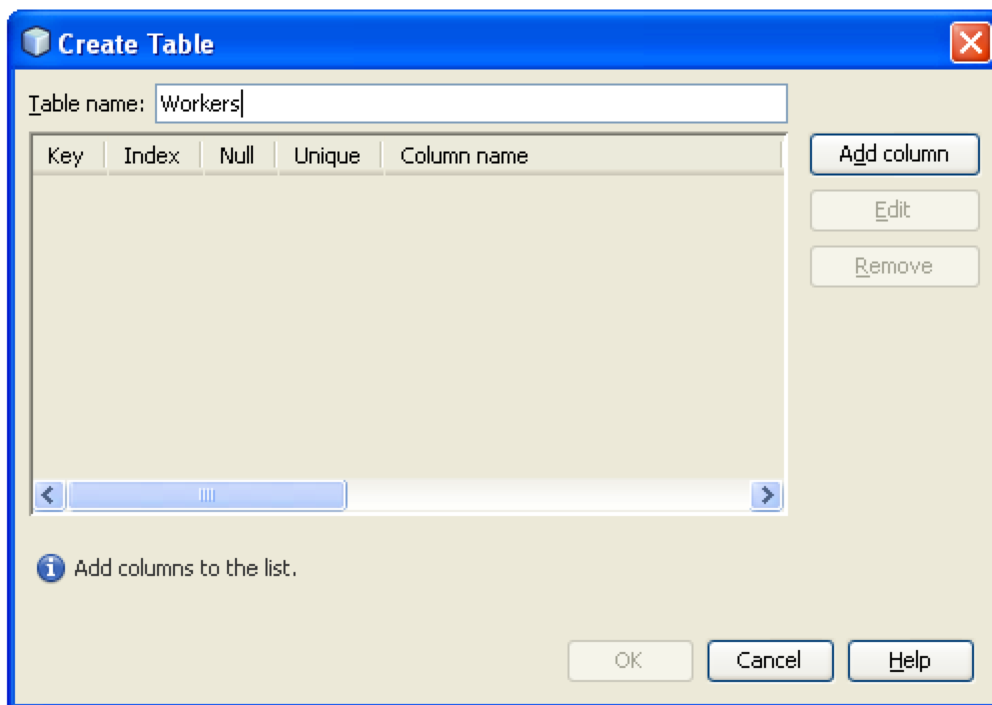
10_من المربع الحوارى الذى سيظهر أدخل كلمة المرور ثم أنقر فوق (ok) بعد ما يتم عملية الاتصال بالخادم ستجد انه قد تم انشاء ثلاث مجلدات و هى Tables , Views, Procedures و حتى تتمكن من رؤية تلك المجلدات فى التويب Serverقم بالنقر دبل كليك على قاعدة البيانات التى انشأناها Employees لتفتح لك بعض المجلدات و لترى المجلدات الثلاث كما فى الشكل التالى:



11_قم بالنقر شمال على المجلد Tables و من القائمة التي تظهر، حدد الامر Create Table كما فى الصورة التالية:



12_عند النقر كما تم شرحه أعلاه يظهر لك المربع الحوارى التالى:



في مربع النص Table Name في الأعلى، أحذف الاسم الافتراضي..Untitled ثم اكتب اسما جديدا للجدول و ليكن مثلا Workers....و هكذا سيكون لديك جدول يسمى Workers

13_إضافة حقول للجدول و يتم ذلك عن طريق النقر فوق زر الامر Add Column فى المربع الحوارى أعلاه ليظهر لك المربع الحوارى التالى:

Add Column

Name: ID

Type: INTEGER

Size: Scale:

Default:

Constraints

Primary key Unique Null Index

Check:

OK Cancel

وهكذا يمكنك إنشاء قاعدة بيانات mysql و جداولها من خلال النت بينز على القرص الصلب

ثالثا : كيفية إنشاء قاعدة بيانات mysql على مقلم بعيد (شبكة الانترنت):

بداية يجب أن تمتلك سرفر و يدعم هذا السرفر قواعد البيانات.... mysql لذلك سنقوم بالتجربة على سرفر مجانى على الرابط التالى:

http://www.db4free.net/d4f_apply.php

قم بالدخول على الرابط أعلاه ستجد أنه يطالبك ببعض البيانات لكى تستطيع إنشاء قاعدة بيانات mysql على هذا السرفر كما فى الصورة التالية:

MySQL database name: mysq20

MySQL username: egytech

MySQL user password: ●●●●●●

MySQL user password verification: ●●●●●●

Allow access from following IP **: %

Default collation ***: utf8_general_ci

Email address: zakyshny@gmail.com

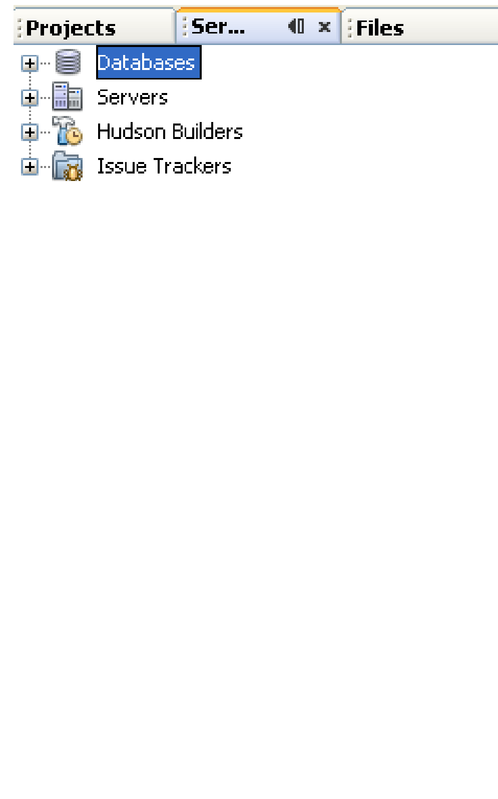
I have read the conditions of use and I agree with them.

Apply for MySQL Account

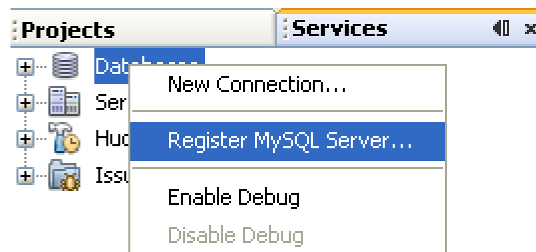
قم بادخال اسم قادة البيانات mysql وفى مثالنا هذا كتبنا... mysq20 بعد ذلك أدخلنا أسم المستخدم... egytech ثم أدخلنا كلمة المرور ....ولا تنسى أن تضع البريد الالكترونى الخاص بك لانه سوف يتم ارسال كود التنشيط عليه .....ادخل على الميل هتلاقى رسالة تأكيد الاشتراك

قم بتأكيد الاشتراك ثم اتبع الخطوات التالية:

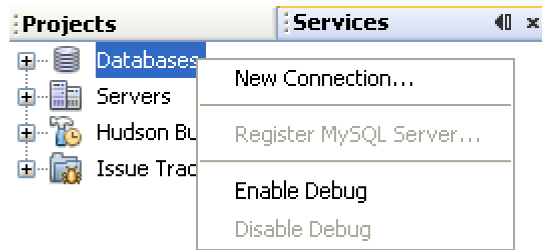
1_قم بفتح NetBeans ومن ثم قم بالضغط على التويب Services فى القوائم الجانيية كما فى الشكل التالى:



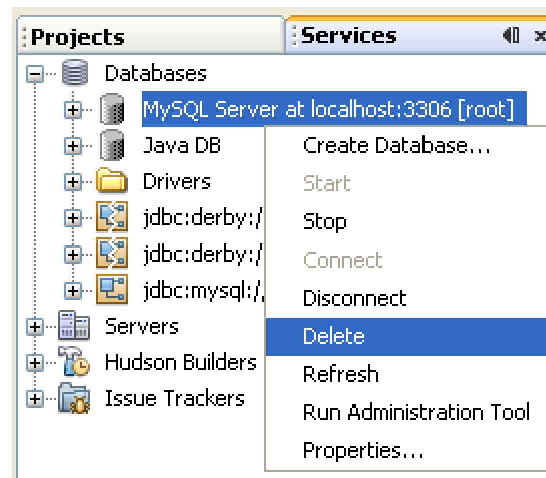
2_قم بتحديد ايقونة Databases ثم كليك يمين ومن القائمة التى سوف تظهر أختر الامر register mysql serviceكما فى الشكل التالى:



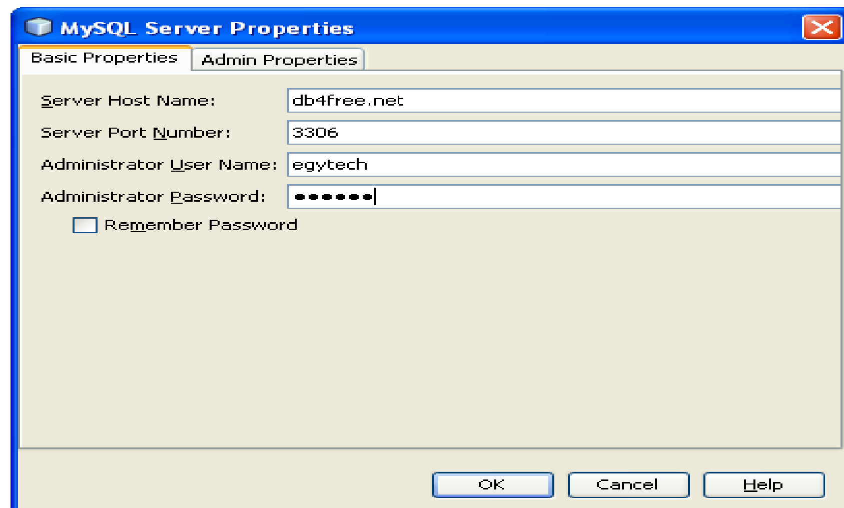
لكن ربما تجد أن الامر register mysql service غير نشط كما فى الصورة التالية:



وهذا معناه أنك قد قمت من قبل بتسجيل معلومات السرفر لذلك قم بحذف معلومات السرفر القديم كما فى الشكل التالى:

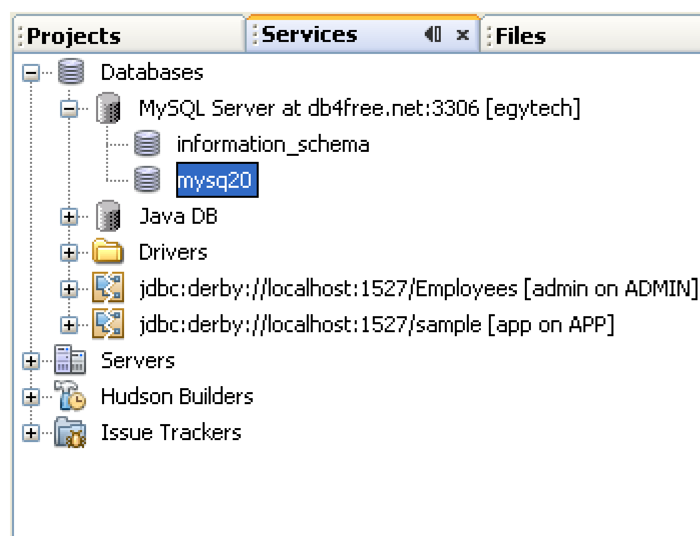


3_عند النقر على الامر register mysql service كما هو ظاهر أعلاه سيظهر لك المربع الحوارى التالى:

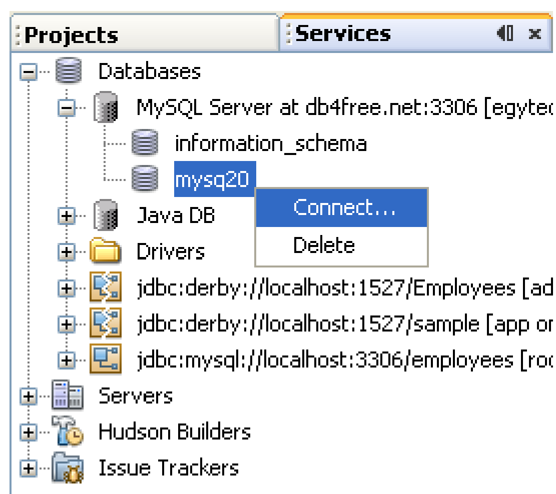


يطالبك هذا المربع service hots name وهو فى مثالنا هذا... db4free.net وايضا رقم المنفذ service portوطبعا هو نفس رقم المنفذ الذى حددناه اثناء تنصيب... service اما اسم المستخدم فهو egytech وهو نفس اسم المستخدم وكلمة المرور التى حددناها اثناء التسجيل فى السرفر ....بعد اتمام ادخال البيانات انقر على زر الامر (موافق).

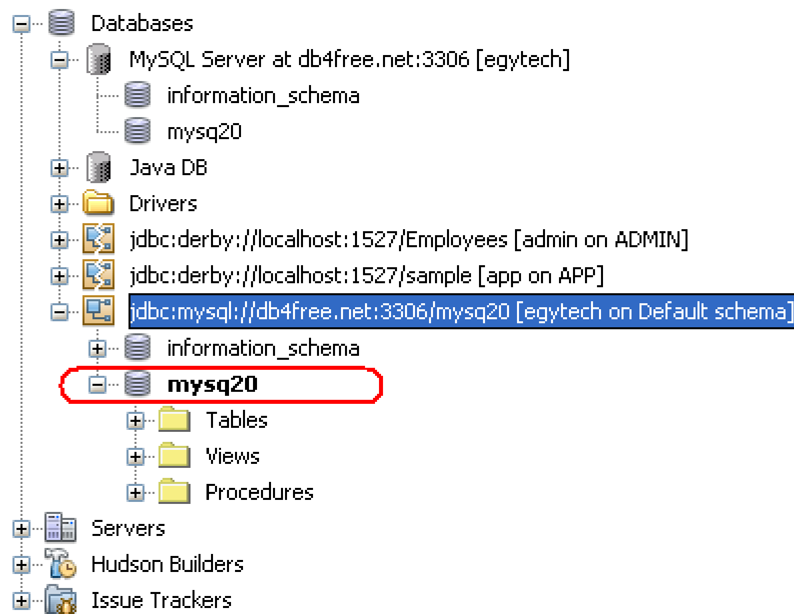
4_بعد ان يتم الاتصال بالسرفر db4free.net ستجد انه قد جلب جميع قواعد البيانات الموجودة عليه كما فى الشكل التالى:



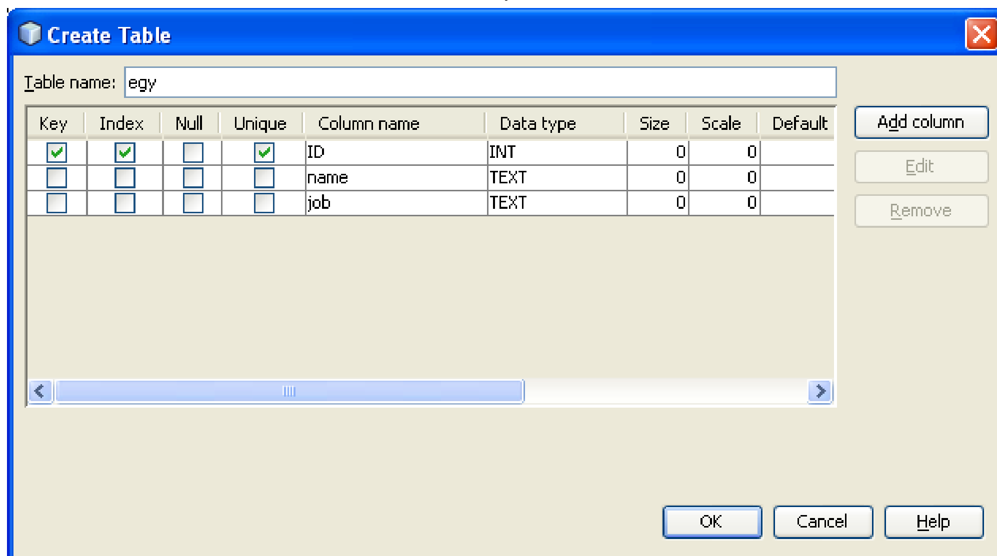
من الصورة أعلاه حدد قاعدة البيانات التى قد قمنا باشائها اثناء التسجيل فى السرفر وهى mysq20 ومن القائمة التى ستظهر أختر الامر connet كما فى الشكل التالى:



5_بعد عملية الاتصال بقاعدة البيانات mysq20 ستجد ايقونة بأسم jdbc:mysql://db4free.net:3306/mysq20 [egytech on Default schema]و عند النقر عليها دبل كليك ستجد قاعدة البيانات التى انشأناها كما فى الشكل التالى :



6_الان قم بالنقر يمين على المجلد Tables و من القائمة التي تظهر، حدد الامر Create Table كما فى الصورة التالية :



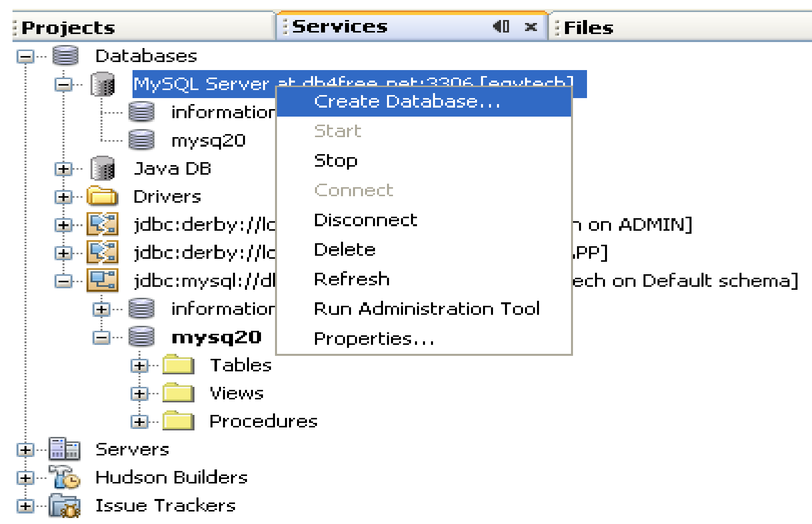
ومن هنا نستطيع انشاء الجدول الخاصة بنا فى قاعدة بيانات mysq20 على سرفر... db4free.net وكما ترون فى الصوره أعلاه قمنا بانشاء الجدول EGY وبه الحقول التالية ID , NAME . JOB و لو فتحت السرفر الخاص بك ستجد الجدول به كما فى الصورة التالية:

جدول البيانات

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	ID	int(11)			No	None	
2	NAME	text	utf8_general_ci		No	None	
3	JOB	text	utf8_general_ci		No	None	

حقول جدول البيانات التى انشأنها بالنت بينز

...فى هذا المثال التجريبى يجب ملاحظة أننا قد أنشأنا قاعدة بيانات mysq20 عن طريق السرفر db4free.net عند التسجيل فية و ليس عن طريق NetBeans و كل ما انشأنها بالنت بينز هو جداول تلك القاعدة ....لكن تستطيع أن تشىء قاعدة بيانات باستخدام NetBeans فقط قم بتحديد السرفر الخاص بك ثم كليك يمين ومن القائمة التى ستظظهر أختر الامر Create Databaseكما فى الصورة التالية:



لكن للاسف لن تستطيع ان تجرب ذلك على هذا السرفر المجانى db4free.net و ذلك لكون لا يتيح الا بانشاء قاعدة بيانات واحد فقط و هى التى أنشاناها عند التسجيل ....لذلك جرب على سرفر أخر

رابعا : الاتصال بقواعد بيانات mysq

تعلمنا كيفية إنشاء قاعدة بيانات MySQL سواء كانت على القرص الصلب أو على شبكة الانترنت ....اليوم بإذن الله تعالى سنتعلم سويا كيفية الاتصال بتلك القاعدة من خلال JAVA وذلك على النحو التالى :

1_قم بإنشاء قاعدة بيانات MySQL و ليكن Employees بها جدول وليكن egy وبه الحقول التالية

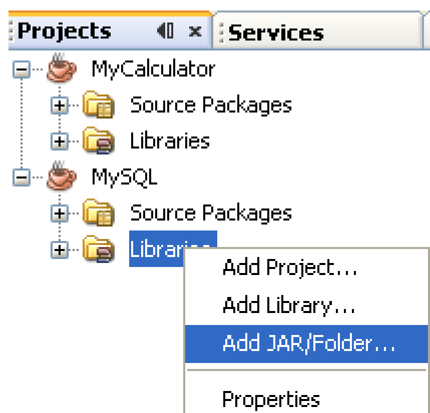
Column Name: ID
Data Type: int

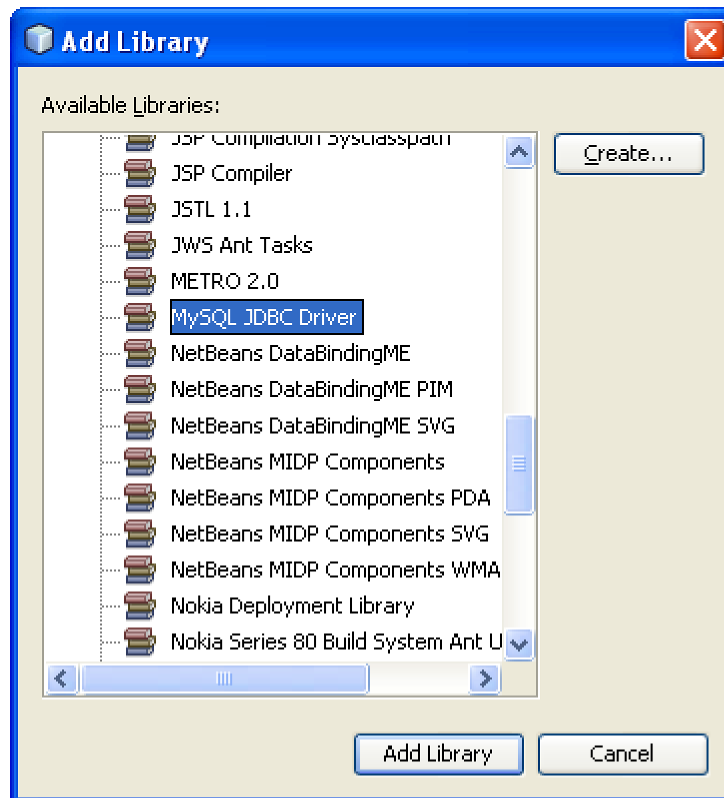
Column Name: Name
Data Type: Text

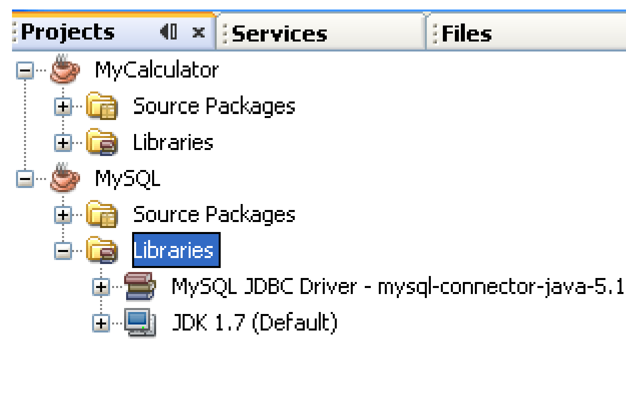
Column Name: Job
Data Type: Text

2_قم بفتح مشروع جديد فى NetBeans و ليكن MySQL مع إزالة علامة (الصح) من خانة create main classوذلك لاننا لانريد الان إنشاء الفئة الرئسية ...قم بإنشاء نافذة رسومية وضع عليها العناصر الرسومية اللازمة كما فى الشكل التالى:

3_قم بادراج المكتبة MySQL JDBC Driver و ذلك عن طريق تحديد ايقونة Library فى القائمة الجانبية للمشروع ومن القائمة التى ستظهر أختر الامر Add Library سيتم فتح لك مربع حوارى للمكتبات الموجودة على جهازك أختر منها MySQL JDBC Driverثم ok لتجد المكتبة قد اضيفتت للمشروعك كما فى الصور التالية:







4_نقوم استدعاء جميع الدوال التابعه للمكتبه **SQL** و ايضا ادراج المكتبة التي تتعامل مع صناديق الحوار JOptionPaneكما تعلمنا سابقا و على النحو التالي:

```
import java.sql.*;
import javax.swing.JOptionPane;
```

5_الاعلان عن الكائنات التي تستخدم لاتمام عملية ربط قاعدة البيانات مع البرنامج باستخدام الكائنات كما تعلمنا سابقا و هي : الكائن Connection و هو الخاص بنص الاتصال... الكائن Statement و هو الخاص بجمل التحكم ....و الكائن Result Setو هو الخاص بمؤشر البيانات وذلك عن طريق الكود التالي:

```
Connection con;
Statement st;
ResultSet rs;
```

6_نقوم بإنشاء طريقة و ليكن مثلا DoConnect كما توعدنا فى الدروس السابقة و ذلك لوضع كافة التعليمات البرمجية الخاصة بالاتصال بقاعدة البيانات وجدولها

7_داخل الطريقة سنقوم بالاعلان عن ثلاث متغيرات وليتم تخزين فيها رابط قاعدة البيانات واسم المستخدم و كلمة المرور و ذلك على النحو التالى:

```
String url = "jdbc:mysql://localhost:3306/employees ";
   String user = "root"
   String password = "123"
```

8_نقوم بأستدعاء الطريقة getConnection الموجودة فى الفصيلة DriverManager من الاتصال بقاعدة بيانات employeesو ذلك عن طريق الكود التالى:

```
con = DriverManager.getConnection(url, user, password);
```

ولكن في حالة عدم القدرة على الاتصال بقاعدة البيانات سيحدث الاستثناء SQLException لذلك سوف يتم معالجه كما فعلنا فى الدرس السابق.

9_نقوم بإستدعاء الطريقة createStatement لايجاء الكائن Statement والذي سيستخدم للاستعلام من قاعدة البيانات وذلك عن طريق الكود التالى:

```
st = con.createStatement();
```

10_نقوم بجلب قيم الحقول فى الجدول و ذلك تنفيذ الاستعلام من خلال استدعاء الطريقة executeQuery الموجودة فى الصنف Statement و الذى قد عبرنا عنه بالكائن rs و ذلك على النحو التالى:

```
rs = st.executeQuery("select * from egy");
```

11_نقوم بإسناد حقول الجدول الى مربعات النصوص فى النافذة عن طريق الكود التالى:

```
rs.next( );
    int id_col = rs.getInt("ID");
    String id =Integer.toString(id_col);
    String name = rs.getString("name");
    String job = rs.getString("job");

    textid.setText(id);
    textname.setText(name);
    textjob.setText(job);
```

و هذا طبعا نفس الكود الذى تم استخدامة فى الدرس السابق ...كما أننا قمنا بالاعلان عن المتغير الرقمى الصحيح id_col و ذلك لتخزين قيمة الحقل (ID) الموجود فى جدول البيانات الذى نوعه ايضا رقمى..(integer) لكن المشكلة ان حقول النص فى الجافا لا تقبل الا قيم نصية بها لذلك قمنا بالاعلان عن المتغير النصى (id) لتخزين فية قيمة المتغير id_colبعد تحويلة الى متغير نصى باستخدام الامر Integer.toString

و فى النهاية فإن الكود النهائى سيكون كما فى الصورة التالية:

```
public void DoConnect( ) {

    String url = "jdbc:mysql://localhost:3306/employees";
    String user = "root";
```

```java
    String password = "123";

try {
        con = DriverManager.getConnection(url, user, password);
        st = con.createStatement();
        rs = st.executeQuery("select * from egy");

    rs.next( );
    int id_col = rs.getInt("ID");
    String id =Integer.toString(id_col);
    String name = rs.getString("name");
    String job = rs.getString("job");

    textid.setText(id);
    textname.setText(name);
    textjob.setText(job);


        } catch (SQLException ex) {
        JOptionPane.showMessageDialog(NewJFrame.this, ex.getMessage());
        }
}
```

12_الان قم بوضع DoConnect فى الدالة الرئيسية للبرنامج وستجد بيانات جدول قاعدة البيانات قد ظهرت فى مربعات النصوص كما فى الشكل التالى :

أما برمجة أزار التنقل و الحذف و الاضافة و خلافة فهى نفس فكرة الدرس السابق

الفصل الثالث : قواعد بيانات Access

مازلنا فى الدروس الخاصة بقواعد البيانات فى الجافا ....ففى الدرس السابق تعلمنا سويا كيفية إنشاء قاعدة بيانات Java DBو كيفية التعامل معها ....اليوم إن شاء الله تعالى سنتعلم سويا ايضا كيفية الاتصال و التعامل مع قاعدة بيانات أكسس و ذلك على النحو التالى:

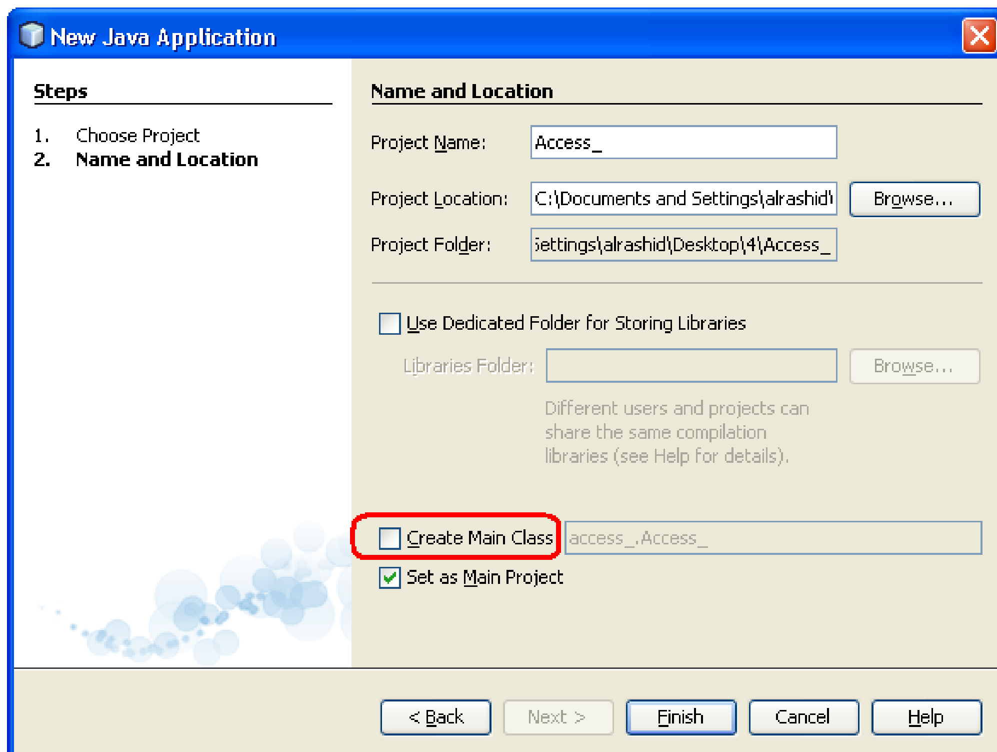
1_قم بإنشاء قاعدة بيانات أكسس و ليكن db1 بها جدول وليكن Workers وبه الحقول التالية

Column Name: ID
Data Type: Number

Column Name: Name
Data Type: Text

Column Name: Job
Data Type: Text

2_قم بفتح مشروع جديد فى NetBeans و ليكن Access مع إزالة علامة (الصح) من خانة create main class وذلك لاننا لانريد الان إنشاء الفئة الرئسية كما فى الشكل التالى:



3_قم بإنشاء نافذة رسومية وضع عليها العناصر الرسومية اللازمة كما فى الشكل التالى:

4_قم بنقل قاعدة البيانات db1 التى قمت بإنشائها الى المجلد الرئيسي للمشروع الذى قمت بإنشائة من خلال

5_نقوم استدعاء جميع الدوال التابعة للمكتبه **SQL** و ايضا ادراج المكتبه التي تتعامل مع صناديق الحوار JOptionPaneكما تعلمنا سابقا و على النحو التالى:

```
import java.sql.*;
import javax.swing.JOptionPane;
```

6_الاعلان عن الكائنات التى تستخدم لاتمام عملية ربط قاعدة البيانات مع البرنامج باستخدام الكائنات كما تعلمنا سابقا و هى : الكائن Connection و هو الخاص بنص الاتصال ...الكائن Statement و هو الخاص بجمل التحكم ....و الكائن Result Setو هو الخاص بمؤشر البيانات وذلك عن طريق الكود التالى:

```
1   import java.sql.*;
2   import javax.swing.JOptionPane;
3
4   public class NewJFrame extends javax.swing.JFrame {
5
6   Connection con;
7   Statement stmt;
8   ResultSet rs;
9
10      public NewJFrame() {
11          initComponents();
12
13
14
15      }
```

ويمكنك ملاحظة أننا قمنا بالاعلان تلك الكائنات على مستوى الفئة حتى نتمكن من استخدامها فى أى حدث من الاحداث داخل الفئة

7_نقوم بإنشاء طريقة و ليكن مثلا DoConnect و ذلك لوضع كافة التعليمات البرمجية الخاصة بالاتصال بقاعدة البيانات وجدولها ....و يتم إنشاء الطريقة باستخدام الكود التالى:

```
public void DoConnect( ) {


}
```

8_نقوم بكتابة الكود التالى لتحميل JDBC Driver الخاص بقواعد البيانات اكسس و ذلك على النحو التالى:

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

كما فى السطر السابق قمن باستدعاء الطريقة forName وذلك لتحميل الفصيلة Class التي تحتوي على مشغل قاعدة البيانات ...و تحميل هذه الفصيلة قد يتسببفي حدوث استثناء في حالة أن الفصيلة المطلوبة لا يمكن تحميلها و هذا الاستثناء هو ClassNotFoundExceptionو بالتالى سنقوم بمعالجة هذا الاستثناء على النحو التالى:

```
try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      } catch (ClassNotFoundException ex) {
       JOptionPane.showMessageDialog( NewJFrame.this, ex.getMessage());
      }
```

9_نقوم بأستدعاء الطريقة getConnection الموجودة فى الفصيلة DriverManager من الاتصال بقاعدة بيانات db1و ذلك عن طريق الكود التالى :

```
con = DriverManager.getConnection("jdbc:odbc:Driver={Microsoft Access Driver
(*.mdb)};DBQ=db1.mdb");
```

يقوم هذا السطر بعملية اتصال بقاعدة البيانات عن طريق نص الاتصال jdbc:odbc:Driver={MicrosoftAccess Driverثم وضعنا أسم قاعدة البيانات db1.mdb ولكن في حالة عدم القدرة على الاتصال بقاعدة البيانات سيحدث الاستثناء SQLException لذلك سوف يتم معالجة كما فعلنا فى الدرس السابق.

10_نقوم بإستدعاء الطريقة createStatement لايجاء الكائن Statement والذي سيستخدم للاستعلام من قاعدة البيانات وذلك عن طريق الكود التالى:

```
stmt=con.createStatement();
```

11_نقوم بالاعلان عن متغير نصى وليكن SQL لحفظ جملة الاستعلام فية و ذلك عن طريق الكود التالى:

```
String SQL = "select * from WORKERS";
```

_ 12تنفيذ الاستعلام من خلال استدعاء الطريقة executeQuery الموجودة فى الصنف Statement و الذى قد عبرنا عنه بالكائن rs و ذلك على النحو التالى:

```
rs = stmt.executeQuery( SQL );
```

13_نقوم بإسناد حقول الجدول الى مربعات النصوص فى النافذة عن طريق الكود التالى:

```
rs.next( );
    int id_col = rs.getInt("ID");
    String id =Integer.toString(id_col);
    String name = rs.getString("name");
    String job = rs.getString("job");

    textid.setText(id);
    textname.setText(name);
    textjob.setText(job);
```

و هذا طبعا نفس الكود الذى تم استخدامة فى الدرس السابق ...فبالنظرة السريعة على الكود سنجد أننا قمنا بالاعلان عن المتغير الرقمى الصحيح id_col و ذلك لتخزين قيمة الحقل (ID) الموجود فى جدول البيانات الذى نوعه ايضا رقمى (integer)....لكن المشكلة ان حقول النص فى الجافا لا تقبل الا قيم نصية بها لذلك قمنا بالاعلان عن المتغير النصى (id)لتخزين فية قيمة المتغير id_col بعد تحويلة الى متغير نصى باستخدام الامر Integer.toString

و فى النهاية فإن الكود النهائى سيكون كما فى الصورة التالية:

```
public void DoConnect( ) {
    try {
                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            } catch (ClassNotFoundException ex) {
               JOptionPane.showMessageDialog( NewJFrame.this, ex.getMessage());}
 try {
con = DriverManager.getConnection("jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=db1.mdb;");
stmt=con.createStatement();

String SQL = "select * from WORKERS";
rs = stmt.executeQuery( SQL );

rs.next( );
       int id_col = rs.getInt("ID");
       String id =Integer.toString(id_col);
       String name = rs.getString("name");
       String job = rs.getString("job");

      textid.setText(id);
      textname.setText(name);
      textjob.setText(job);
}
catch (SQLException err) {
       JOptionPane.showMessageDialog( NewJFrame.this, err.getMessage());
      }
}
```

14_الان قم بوضع DoConnect فى الدالة الرئيسية للبرنامج وستجد بيانات جدول قاعدة البيانات قد ظهرت فى مربعات النصوص كما فى الشكل التالى:

رقم الموظف 1
الاسم ahmed
الوظيفة acont
السجل الاول | التالى | السابق | السجل الاخير
جديد | حفظ | حذف

أما برمجة أزار التنقل و الحذف و الاضافة و خلافة فهى نفس فكرة الدرس السابق

الفصل الرابع : قواعد بيانات Microsoft SQL Server

درس اليوم إن شاء الله تعالى عن كيفية الاتصال و التعامل مع قواعد بيانات Microsoft SQL Server من خلال تطبيقات الجافا وذلك على النحو التالى:

1_قم بالنقر هنـــــــا وذلك للحصول على Microsoft JDBC Driver 4.0 for SQL Server أو سائق الاتصال بقواعد بيانات SQL Server و التى تدعم الاصدارات ٢٠٠٥، ٢٠٠٨ 2012 , R2 2008,

2_قم بتشغيل الملف sqljdbc. EXE الذى تم تحميله ليتم استخراج الملفات إلى الدليل المحدد أو (الافتراضي و هو Microsoft Microsoft JDBC Driver 4.0 for SQL Server). وسوف تجد ملفين بصيغة jar هما sqljdbc.jar (لJDBC 3.0 و) sqljdbc4.jar (لJDBC 4.0)، بالإضافة إلى بعض ملفات DLL وملفات تعليمات.HTML

3_قم باضافة الملف sqljdbc.jar الى مشروعك إذا كنت تستخدم نسخة الجافا JDK 5.0 أو ملف sqljdbc4.jar إذا كنت تستخدم JDK 6.0 أو أى اصدار أعلى....وذلك من خلال النقر يمين على المجلد Libraries ومن القائمة التى ستظهر أختر الامر Add Jar/Folder

4_كتابة الشفرات للازمة على النحو التالى:

أ_ إستيراد المكتبات اللازمة للعمل على النحو التالى:

```
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.SQLException;
```

ب_ نص الاتصال بقاعدة البيانات يكون عن طريق الكود التالى :

```
DriverManager.registerDriver(new com.microsoft.sqlserver.jdbc.SQLServerDriver());
```

أو النص التالى:

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
```

ومع ذلك، هذا ليس مطلوبا منذ إصدار الجافا (JDK 6.0) مع السائق JDBC 4.0 لانه يمكن الكشف عنه وتحميله تلقائيا طالما أن الملف sqljdbc4.jar موجود في مجلد المشروع.

ج_ بناء جملة URL لقاعدة بيانات SQL Server و الاتصال بها كما يلى:

```
String dbURL = "jdbc:sqlserver://localhost\\sqlexpress;user=sa;password=secret";
Connection conn = DriverManager.getConnection(dbURL);
if (conn != null) {
   System.out.println("Connected");
}
```

_حيث أن:

- localhost _أسم المضيف أو عنوان ip للمقلم الذى عليه قاعدة البيانات
- sqlexpress_قاعدة بيانات sql
- user_أسم المستخدم للقاعدة

password_كلمة المرور للقاعدة

وكما ترى فى الشفرة أعلاه لعمل اتصال بالقاعدة نقوم بأستدعاء الطريقة getConnection الموجودة فى الفصيلة DriverManageمع تمرير رابط... url كما يمكنك استخدامه على النحو التالى:

```
String dbURL = "jdbc:sqlserver://localhost\\sqlexpress";
String user = "sa";
String pass = "secret";
conn = DriverManager.getConnection(dbURL, user, pass);
```

و ايضا يمكننا استخدام كائن java.util.Properties لتخزين خصائص الاتصال، كما في المثال التالي:

```
String dbURL = "jdbc:sqlserver://localhost\\sqlexpress";
Properties properties = new Properties();
properties.put("user", "sa");
properties.put("password", "secret");
conn = DriverManager.getConnection(dbURL, properties);
```

فأختر الطريقة التى تناسبك ....على اى حال اليك الكود الكامل للاتصال بقاعدة بيانات SQL Server من خلال تطبيقات الجافا على النحو التالى:

```
import java.sql.Connection;
import java.sql.DatabaseMetaData;
import java.sql.DriverManager;
import java.sql.SQLException;

public class JdbcSQLServerConnection {

  public static void main(String[] args) {

      Connection conn = null;

      try {

         String dbURL = "jdbc:sqlserver://localhost\\sqlexpress";
          String user = "sa";
          String pass = "secret";
          conn = DriverManager.getConnection(dbURL, user, pass);
          if (conn != null) {
```

```
            DatabaseMetaData dm = (DatabaseMetaData) conn.getMetaData();
            System.out.println("Driver name: " + dm.getDriverName());
            System.out.println("Driver version: " + dm.getDriverVersion());
            System.out.println("Product name: " + dm.getDatabaseProductName());
            System.out.println("Product version: " + dm.getDatabaseProductVersion());
        }

    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        try {
            if (conn != null && !conn.isClosed()) {
                conn.close();
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
  }
}
```

...هذا عن الاتصال بقاعدة البيانات ....أما من حيث التعامل معها كالاضافة و الحذف و التعديل و التنقل بين السجلات فهى نفس فكرة الدرس التالى

الباب الرابع : برمجة تطبيقات الجوال

الفصل الاول : إنشاء المشاريع Building MIDlets

أهلا وسهلا أخوانى الكرام و مرحبا بكم فى منتدى مصر التقنى ...مازلنا مع دروة الجافا الرسومية و قد تحدثنا فى أول درس من دروس تلك الدورة ان هناك ثلاث إصدارات الجافا وهى:

- اصدار الإنتربرايس Java Enterprise Edition وهو الخاص بتطبيقات الويب و يختصر الىjava EE
- الاصدار القياسي Java Standard Edition وهى الخاصة بتطبيقات سطح المكتب وهى التى درسناها من بداية الدرس الثانى و حتى نهاية الدرس السابع عشر وتختصر الىjava se
- اصدار المايكرو Java Micro Edition و يختصر java me وقولنا انه هذا الاصدار خاص ببرمجة تطبيقات الجوال والأجهزة المحدودة في المصادرو هو الذى سنقوم بدراسته ابتداء من هذا الدرس إن شاء الله تعالى
-

وقبل أن نبدأ ببرمجة تطبيقات الجوال باستخدام java me أحب أن انوه أن الاساسيات التى درسناها فى java seمن بداية الدرس الثانى وحتى الدرس السابع هى نفسها التى سيتم استخدامها فى تلك التطبيقات ....مثل الاعلان عن الثوابت و المتغيرات و انشاء المصفوفات و الجملة الشرطية وحلقات التكرار ستكون بنفس القواعد التى **درسناها فى JAVA SE**

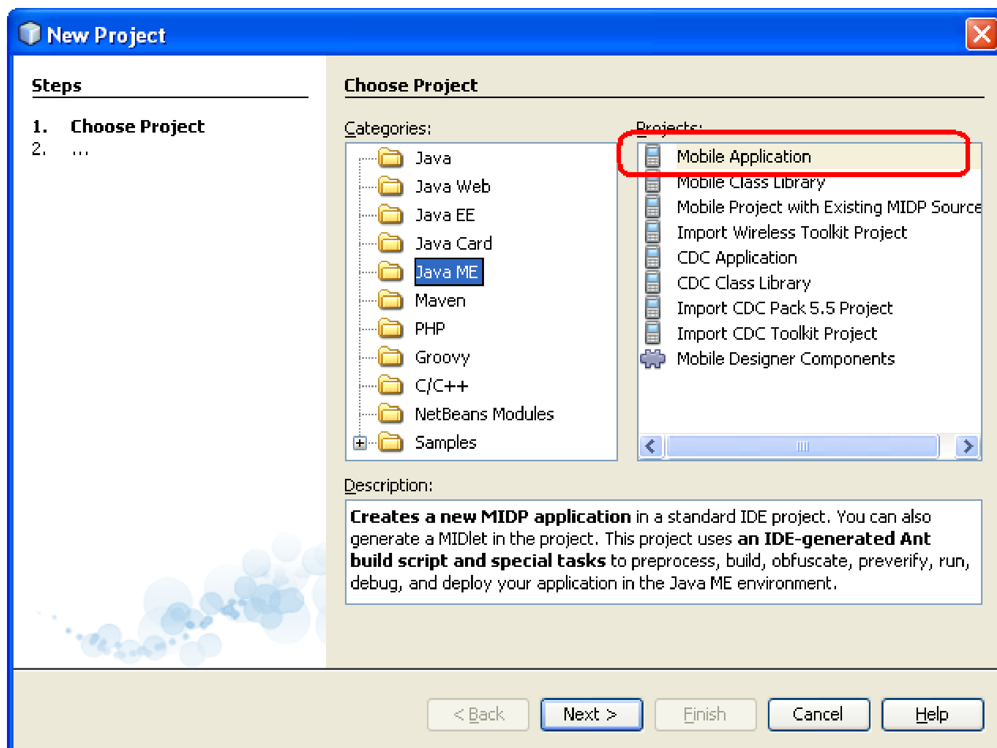
<u>اولا :مفهومMIDlet</u>

تسمى المشاريع القياسية لتطبيقات الجوال MIDlet وهى اختصار للجملة (Mobile Information Device Profile)وهى تطبيقات تستخدم معلومات الملف الشخصي الجهاز المحمول من مواصفات منصة java me التى تحتوى على المكتبات اللازمة لدعم برمجيات الجافا على الجوال ...تلك المنصة تسمى (CLDC) وهى اختصار للجملة الانجليزية... Connected Limited Device Configuration

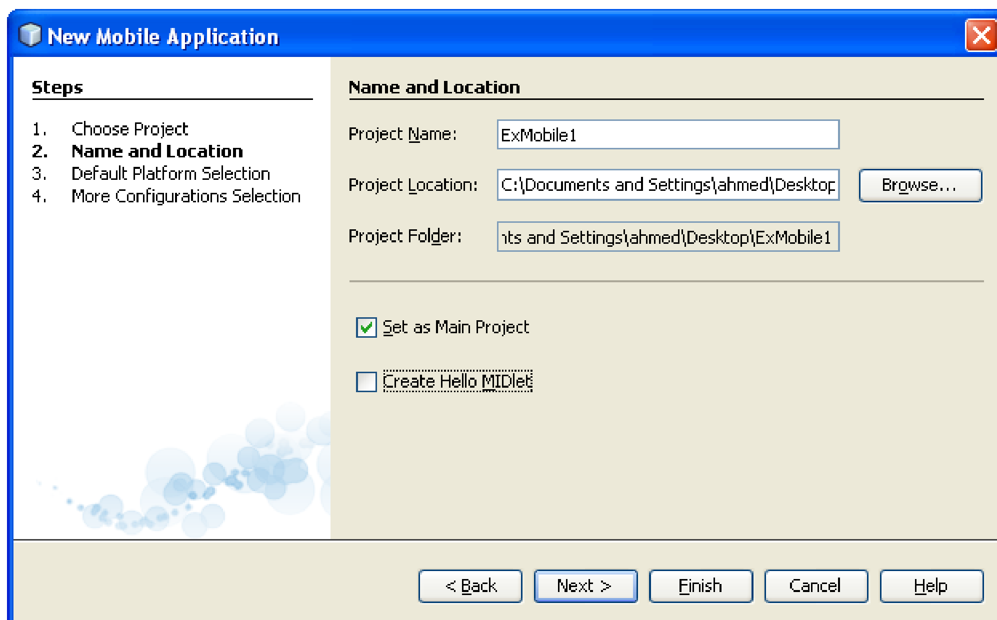
<u>ثانيا : خطوات إنشاء تطبيقات الجوال</u>

1_قم بفتح... NetBeans بعد ذلك قم بالنقر على ايقونة New Project ليتم فتح معالج إنشاء مشروع جديد ....من هذا المعالج ومن التويب (category) أختر JAVA ME ومن التويب project أختر الامر Mobile Application كما فى الصورة التالية:

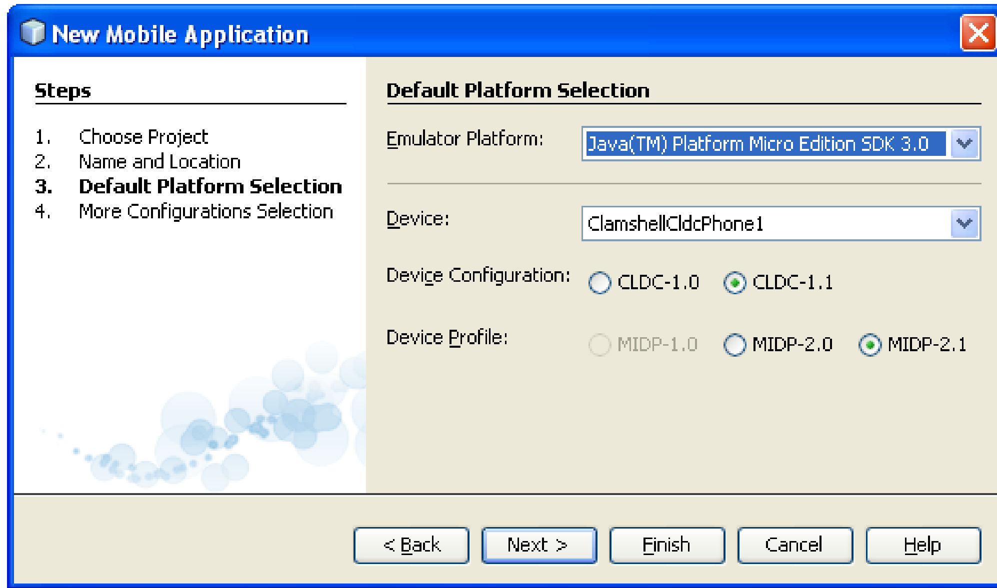
وكما هو ملاحظ يتطلب المربع الحوارى السابق ...اسم للمشروع و مسار حفظ المشروع بالاضافة الى ذلك يوجد خيارين ....الاول set as main project و ذلك لجعل مشروعك رئيسى فى حالة وجود عدة مشاريع اخرى ...الخيار الثانى Create Hello MIDletوهذا لانشاء تطبيق MIDlet جاهز باسم... Hello لذلك سنقوم بإالغاء تحديد هذا الخيار و ذلك لانشاء التطبيق بانفسا ليتثنى لنا معرفة خطوات إنشاء مشروع قياسى MIDlet



2_بعد النقر على زر (NEXT) يظهر لك المربع الحوارى التالى:



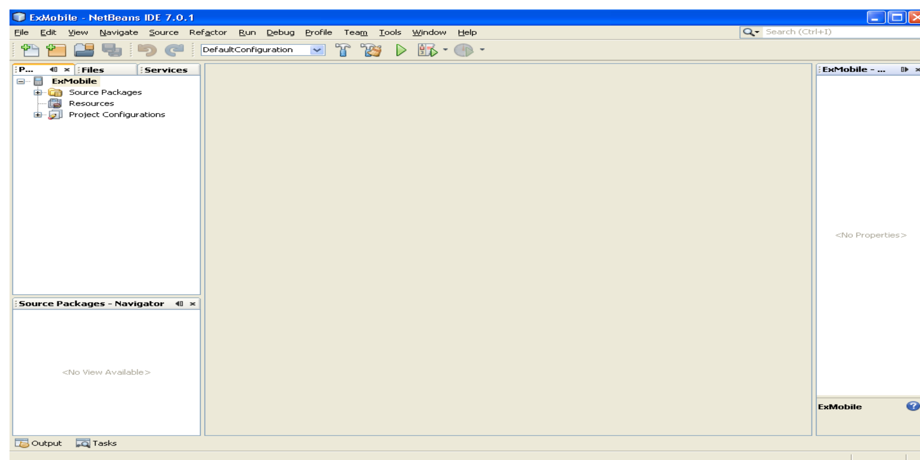
3_بعد النقر على زر الامر (next) يظهر المربع Default Platform Selection و ذلك لتحديد المعلومات الافتراضية اللازمة لانشاء تطبيق الجوال وهو كما فى الصورة التالية:



وكما هو ظاهر فى الصورة السابقة يتطلب المربع الحوارى المعلومات التالية:

- **device** _وهو لاختيار المحاكى الافتراضى الذى سوف تستخدمه لتشغيل عليه تطبيقاتك قبل نقلها الى الجوال ....ويوجد العديد من المحاكى و التى تختلف عن بعضها من حيث) شاشة احادية اللون او شاشة عديدة الالوان وموضع الازرار فى لوحة المفاتيح ....الخ. إذا كنت ترغب في اختبار التطبيق في بيئة معينة (على جهاز محاكى محددة) قد توفير الموارد المتاحة للاستخدام من قبل الشركة المصنعة؛ على سبيل المثال لأجهزة نوكيا، توفر لك device من أجيال مختلفة ونماذج هي forum.nokia.com؛
- **Device Configuration**_وهو الخاص بمعلومات تكوين الجهاز ولحد الان توجد اصدارين التى تدعمها الجوالات هما CLDC-1.0 و.... CLDC-1.1 سنقوم بأختيار الاصدار الاول فهو الاكثر انتشارا فى اجهزة الجوال
- **device profile**_وهو معلومات الملف الشخصي لاجهزة الجوال و هو أمتداد CLDC و سنتختار الاصدار MIDP-2.0 و السبب فى ذلك أن حوالى ٧٥ % من أجهزة الجوال التى صنعت بعد عام ٢٠٠٥ تدعم هذا الاصدار ...

4_بعد الانتهاء من تحديد كافة المعلومات اللازمة يمنكنك النقر على زر الامر(finsh) ليظهر لك بنية netbeans كما فى الشكل التالى :



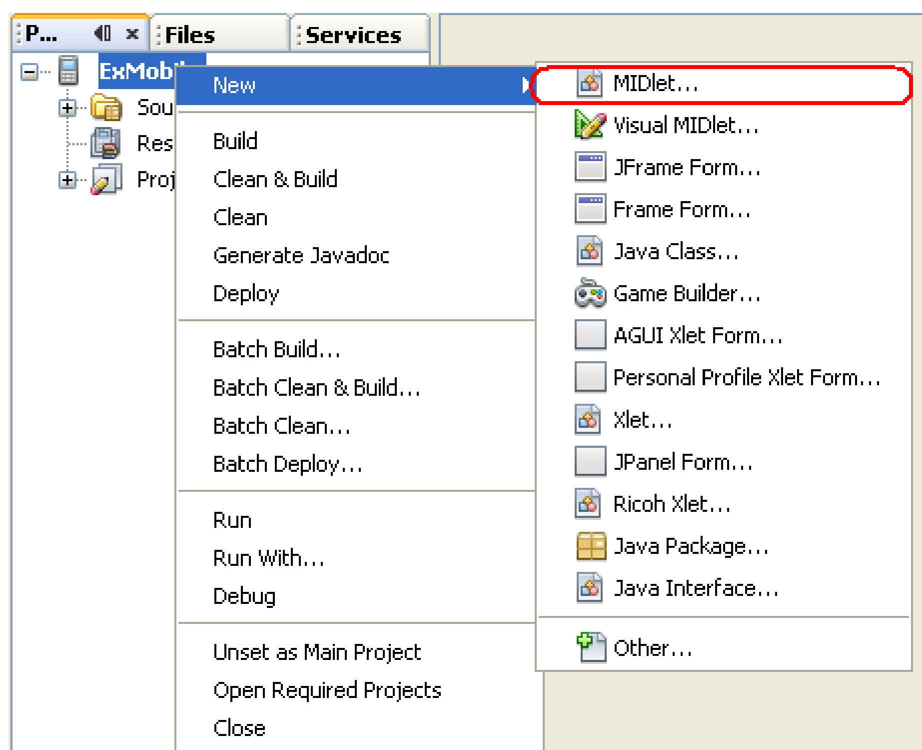
5_كما هو ظاهر فى الصورة أعلاه حتى الان لم يتم الانتهاء من إنشاء المشروع ....ولكى يتم اكتمال إنشاء المشروع يجب علينا تحديد السياق التي على أساسها سوف نبني التطبيق. وهذا يعني إدراج مورد أو ملف من نوع MIDlet ويوجد طريقتين لادراج هذا الملف وهما:

الطريقة الاولى : وهو ادراج المورد أو الملف MIDlet فى بيئه العمل غير رسومية و التى تتيح لنا التعامل مع التطبيق من خلال الشفرة فقط

الطريقة الثانية : وهو ادراج المورد أو الملف MIDlet فى بيئة العمل الرسومية و التى تتيح لنا إنشاء التطبيق من خلال العناصر الرسومية كإضافه النوافذ و القوائم عن طريق السحب و الافلات

بناء المشاريع فى بيئة العمل غير الرسومية

تعالوا الان نتعرف على **الطريقة الاولى** أولا وبعد ذلك سنتعامل مع البيئة الرسومية باقى دروس الدورة لكى نستطيع إدراج المورد أو الملف MIDlet فى البيئة غير الرسومية نقوم بتحديد ايقونة التطبيق ثم كليك يمين ومن القائمة التى ستظهر أختر الامر New ثم الامر MIDlet كما فى الصورة التالية:



سيظهر لك المربع الحوارى التالى و الذى يطالبك ببعض المعلومات كأسم الملف MIDlet و اسم الكلاس الخاص بـMIDlet ويمكنك تغير البيانات الافتراضية أو تركها كما هى إن شئت ثم قم بالنقر على زر الامر (finsh) ليتم فتح صفحة كتابة الشفرة كما فى الصور التالية:

```
1
2  import javax.microedition.midlet.*;
3
4
5  public class Midlet extends MIDlet {
6
7      public void startApp() {
8      }
9
10     public void pauseApp() {
11     }
12
13     public void destroyApp(boolean unconditional) {
14     }
15 }
```

يلاحظ فى الكود أعلاه أنه تم انشاء الفئة أو الكلاس Midlet المشتق من المودر Midlet وبداخل هذا الكلاس ثلاث طرق وهى:

- **startApp**_ وتستخدم هذه الطريقة لتشغيل التطبيق أو بمعنى أوضح نقطة البداية للتطبيق.
- **pauseApp**_وتستخدم هذه الطريقة لتنفيذ وظيفة ما في حالة وقوع حدث يؤدى الى عرقلة تشغيل التطبيق مثل تلقي مكالمة هاتفية أثناء تشغيل التطبيق
- **destroyApp**_هى الطريقة المستخدمة لإغلاق التطبيق؛ يتم تنفيذه في نهاية التطبيق.
-

تعالوا الان أخوانى الكرام نقوم بعمل تطبيق جوال بناء على الملعومات السابقة لاظهار نص مكتوب فيه "مرحبا بكم فى منتدى مصر التقنى"

لعرض نص أو رسالة على شاشة الجوال لابد من استخدام الكائن Display و الذى يقوم بتهئة المورد MIDlet فى بداية التطبيق.... هذا الكائن Display هو أحد الكائنات التابعة للمكتبة lcdui لذلك سيتم استيرادها ....وبالتالى سيكون الكود على النحو التالى :

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Midlet extends MIDlet {
private Display display = null;

public void startApp() {

if(display==null)
display = Display.getDisplay(this);
}
public void pauseApp() {
}
public void destroyApp(boolean unconditional) {
}
}
```

وكما تلاحظون فى الكود السابق ....ففى السطر الثانى قمنا باستيراد المكتبة.... lcdui وفى السطر الرابع قمنا بإنشاء الكائن....Displayفى طريقة startApp استخدمنا جملة شرطية للتأكد أن الكائن display يسلوى null فإن كان الامر كذلك يتم تهيئة المورد من خلال الامر getDisplay(this) وكلمة this تعنى هذا المورد

_الكود السابق حتى الان ليس كافيا لعرض الرسالة وذلك لاننا نحتاج الى وعاء للرسالة وليكن مربع نص وهى احد العناصر الرسومية للجافا لتطبيقات الجافا و هى تختلف عن مربعات النصوص المستخدمة فى تطبيقات سطح المكتب ....ويتم إضافة مربع النص باستخدام الكود التالى :

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class Midlet extends MIDlet {
 private Display display = null;
 private TextBox textmass;

   public void startApp() {

   if(display==null)
        display = Display.getDisplay(this);
   textmass=new TextBox("أول تطبيق جوال","مرحبا بكم فى منتدى مصر التقنى",١٠٠،٠);
   display.setCurrent(textmass);

   }

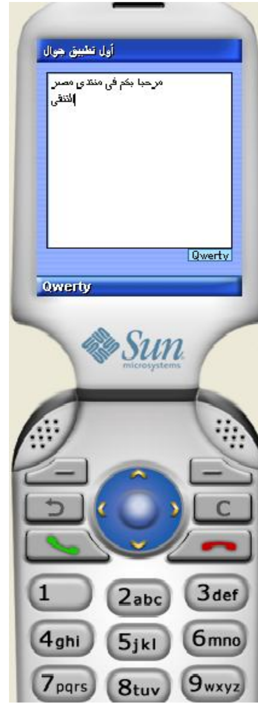
   public void pauseApp() {
   }

   public void destroyApp(boolean unconditional) {
   }
}
```

وكما تلاحظون الاكواد الجديدة فى الكود السابق عما قبله ....ففى السطر الخامس قمنا بالاعلان عن مربع نص وتم تسميته textmassوففى السطر التاسع قمنا اشتقاق كائن جديد من مربع النص و اسناده الى textmass مع وضع البيانات للازمة لاظهار الرسالة ....أما فى السطر العاشر قمنا باستخدام الطريقة setCurrent للكائن display لتشغيل و إظهار مربع النص ....الان قم بالنقر على زر التشغيل فى شريط أدوات netbeans



ليظهر لك المحاكى وبه رسالت الترحيب كما فى الصورة التالية :



و يمكنك ايضا النقر على لوحة مفاتيح جهاز المحاكى على زر (رجوع) لترى ايقونة تطبيقك كما فى الشكل التالى



رغم بساطة التطبيق فهو قد فهمنا من خلاله كيفية إنشاء تطبيق جوال من خلال البنيية غير الرسومية وهكذا نكون قد أنتهينا من هذه الطريقة فى انشاء المشاريع و الى اللقاء إن شاء الله تعالى فى شرح الطريقة الاخرى الرسومية لانشاء تطبيقات الجافا و ايضا دورس أخرى إن شاء الله تعالى

بناء المشاريع فى بيئة العمل الرسومية

إخوانى الكرام أهلا وسهلا بكم ....فى الدرس السابق تحدثنا عن كيفية إنشاء تطبيقات الجوال MIDlet فى البنية غير الرسومية وقمنا بعمل تطبيق بسيط باستخدام الشفرة فقط لعرض رسالة ترحيب ....فى هذا الدرس إن شاء الله تعالى سنتعلم سويا كيفية إنشاء تلك التطبيقات فى البنية الرسومية و ذلك عن النحو التالى:

أولا : خطوات إنشاء تطبيقات الجوال فى البنية الرسومية

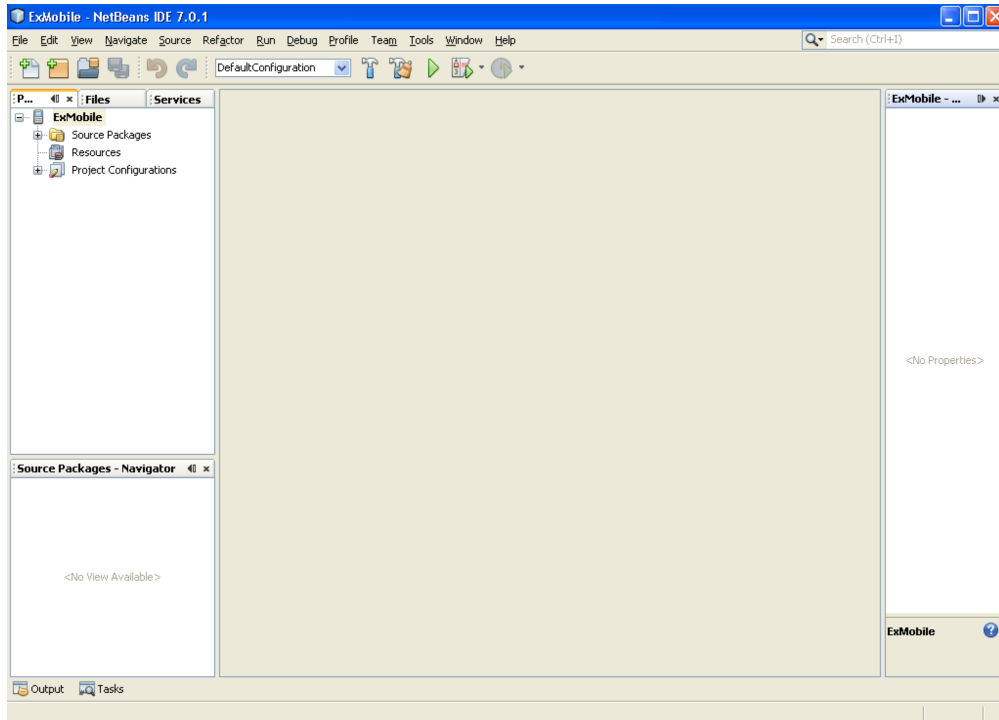
1_ قم: بفتح... NetBeans بعد ذلك قم بالنقر على ايقونة New Project ليتم فتح معالج إنشاء مشروع جديد ....من هذا المعالج ومن التويب (category) أختر JAVA ME ومن التويب project أختر الامر Mobile Applicationثم قم بالنقر على زر الامر Next

2_سيظهر لك مربع حوارى يطالبك .باسم للمشروع و مسار حفظ المشروع بالاضافة الى ذلك يوجد خيارين ....الاول set as main project و ذلك لجعل مشروعك رئيسى فى حالة وجود عدة مشاريع اخرى ...الخيار الثانى Create Hello MIDlet وهذا لانشاء تطبيق MIDlet جاهز باسم... Hello لذلك سنقوم بإالغاء تحديد هذا الخيار و ذلك لانشاء التطبيق بانفسا ....ثم قم بالنقر على زر الامر Next

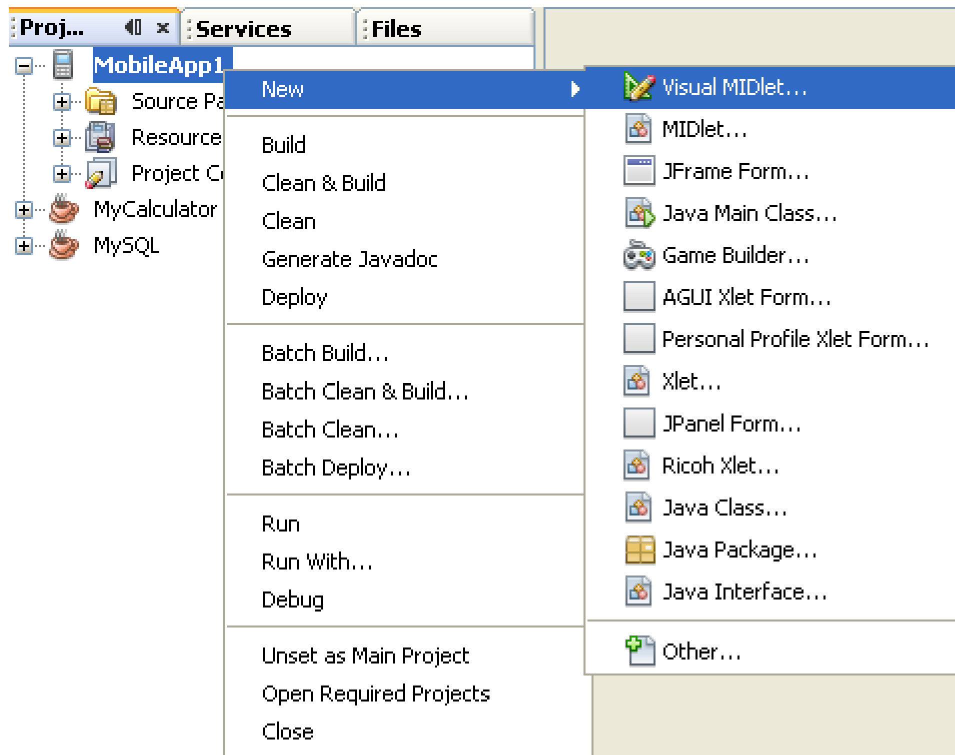
3_بعد النقر على زر الامرNextيظهر المربع Default Platform Selection و ذلك لتحديد المعلومات الافتراضية اللازمة لانشاء تطبيق الجوال و التى قد تحدثنا عنها فى الدرس السابق مثل

device , Device Configuration , device profile

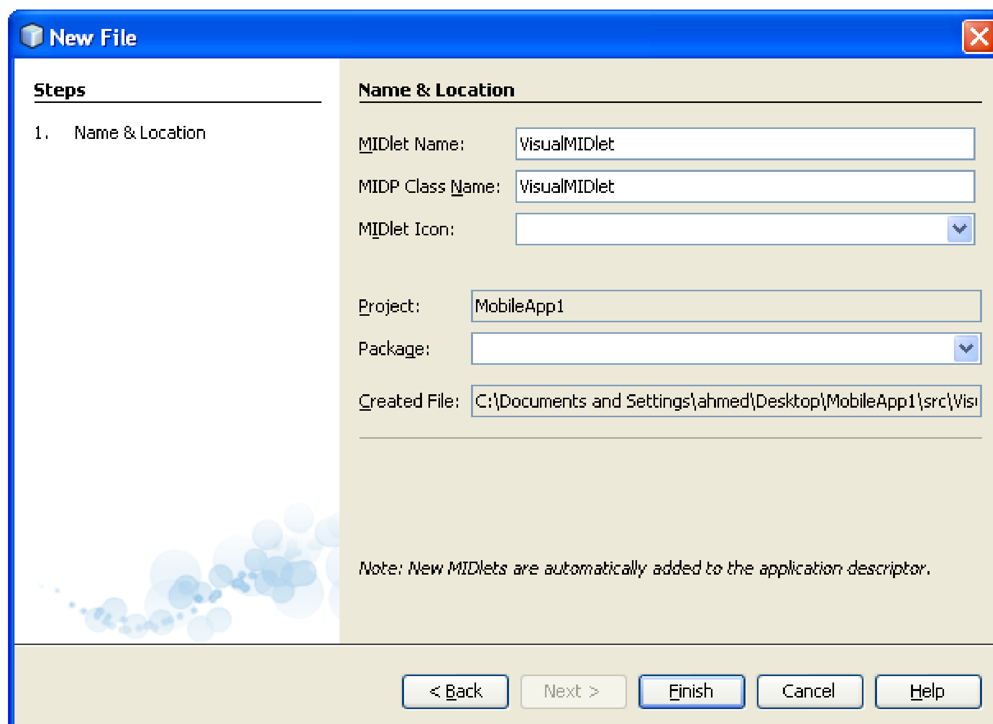
4_بعد الانتهاء من تحديد كافة المعلومات اللازمة يمنكنك النقر على زر الامر(finsh) ليظهر لك بنية netbeans كما فى الشكل التالى:



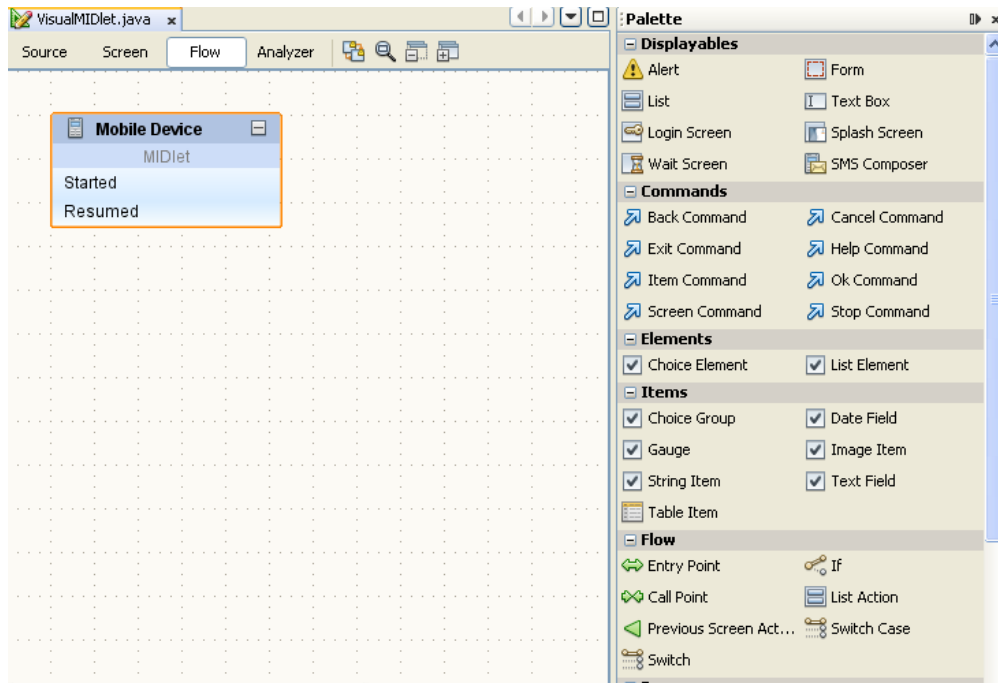
5_نقوم بإدراج المورد أو الملف MIDlet فى البيئة الرسومية و ذلك عن طريق تحديد ايقونة التطبيق ثم كليك يمين ومن القائمة التى ستظهر أختر الامر New ثم الامر Visual MIDlet كما فى الصورة التالية:



6_ سيظهر لك المربع الحوارى التالى و الذى يطالبك ببعض المعلومات كأسم الملف MIDlet و اسم الكلاس الخاص ب MIDletويمكنك تغير البيانات الافتراضية أو تركها كما هى إن شئت كما فى الصورة التالية:

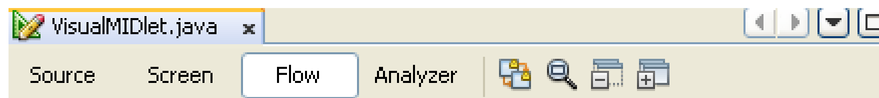


7_بعد الانتهاء من تحديد كافة المعلومات اللازمة للمربع الحوارى السابق قم بالنقر على زر الامر (next) لتظهر لك البنية الرسومية كما فى الشكل التالى:



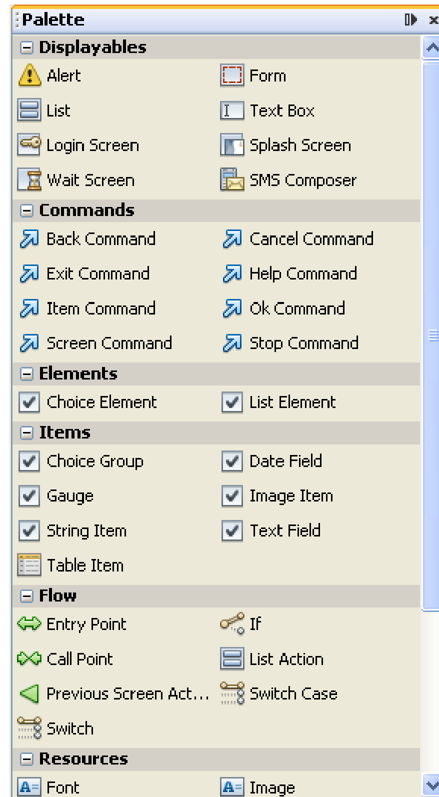
وكما ترى أخى الكريم فى الصورة أعلاه ستجد ملف MIDlet كمخطط بيانى فى شاشة Flow و تعالوا الان الاخوة الافاضل نتعرف على تلك البنية الرسومية:

أولا : يحتوى شريط التويب العلوى الذى فى الصورة التالية على اربعة خيارات للتعامل مع التطبيق وهى



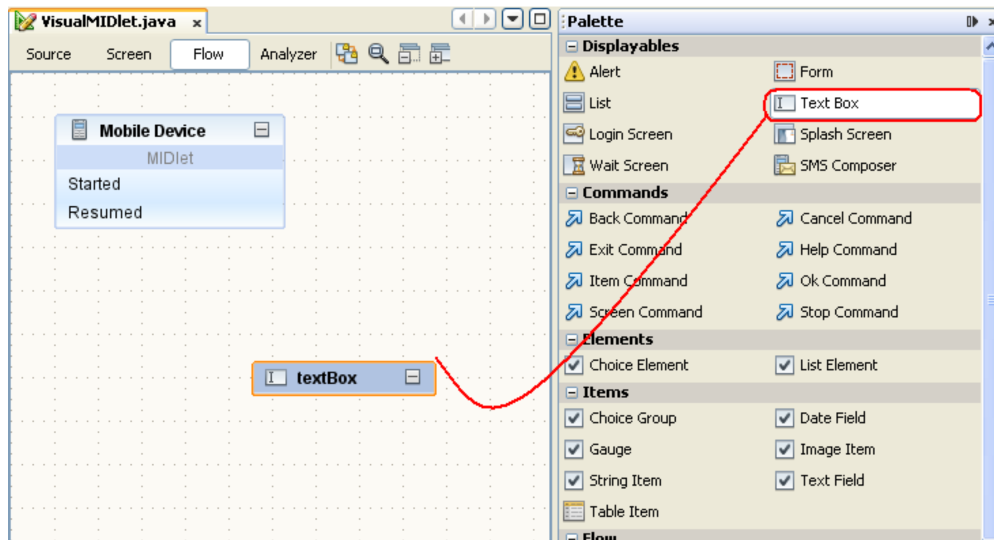
- **Source** وهذا التويب عند النقر عليه يفتح صفحة كتابة الشفرة و ذلك لكتابة اى كود او شفرة تريد تنفيذها
- **Screan** وهذا التويب عند النقر عليه يفتح واجهة رسومية تحاكى وجهة الجوال لتستطيع بعدها إدراج العناصر الرسومية عليها عن طريق السحب و الافلات.
- **Flow** وهذا التويب عند النقر عليه يفتح صفحة كمخطط للعناصر الرسومية و اوامر الربط بينهما و ايضا يتح لنا ادارج الكائنات التابعة للفئة Displayable للتطبيق عن طريق السحب و الافلات
- **analyzer** وهذا التويب عند النقر علية يفتح صفحة تحليل لجميع العناصر و الموارد و الملفات التى تم إدراجها للتطبيق

ثانيا : فى الجهة اليمنى ستجد القائمة palette و التى تحتوى على الكثير من المكونات الرسومية و أوامر التنفيذ command التى يمكنك استخدامها فى تطبيقك منها ما ينتمى الى الفئة Displayable ومنها ما ينتمى الى الفئة Item كما فى الصورة التالية:



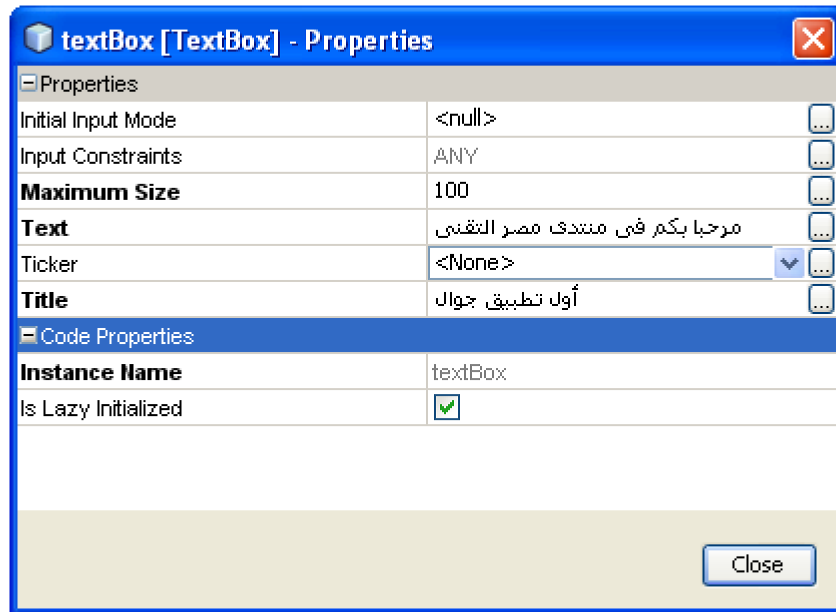
و سيتم شرح تلك المكونات كلا على حد فى دروس متقدمة ....الان وبعد ما تعرفنا على تلك البنية الرسومية تعالوا نقوم بانشاء نفس التطبيق السابق الذى تم شرحة فى البنيه غير الرسومية وهو عبارة عن تطبيق لاظهار رسالة ترحيب ....ويتم ذلك عن طريق الخطوات التالية:

1_ قم بإدراج المكون textbox الى المشروع و ذلك عن طريق سحب المكون من القائمة palette ثم إفلاته الى المشروع من خلال التويب Flow كما فى الصورة التالية:

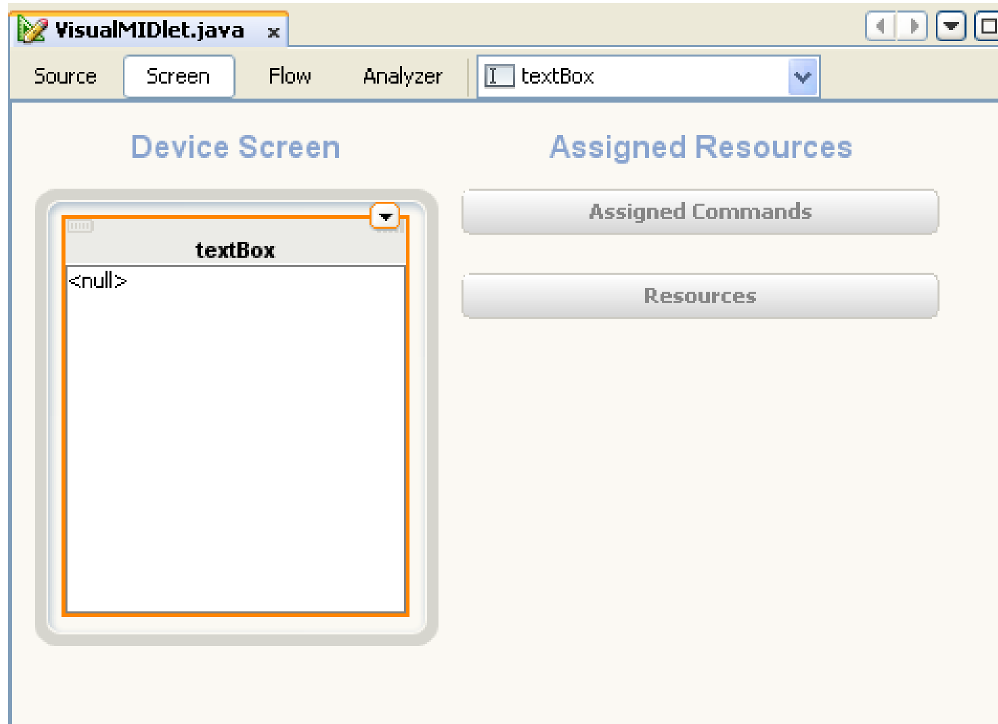


ملحوظة / جميع الكائنات التى تنتمى للفئة Displayable لا يمكن إدراجها للمشروع عن طريق السحب و الافلات إلا من خلال التويب Flow

2_قم بتحديد المكون فى التويب ثم كليك يمين ومن القائمة التى ستظهر أختر الامر propertites لتظهر لك نافذة الخصائص كما فى الشكل التالى:



ومن نافذة قمنا بكتابة رسالة الترحيب المراد عرضها من خلال الخاصية text وايضا قمنا بوضع عنوان للرسالة من خلال الخاصية Titel بعد ذلك قمنا بإغلاق نافذة الخصائص..... ويمكنك فعل كل ذلك من خلال التويب Screan والذى يظهر لك العنصر الرسومى على محاكى الجوال بالشكل التالى:



بقم النقر المزدوج على عنوان المكون textbox الموجود اعلى العنصر و الذى يأخذ القيمة textbox لتسطيع

كتابة عنوان للعنصر وبعد ما يتم الكتابة قم بالنقر على زر الادخال (Enter) كما فى الصورة التالية:

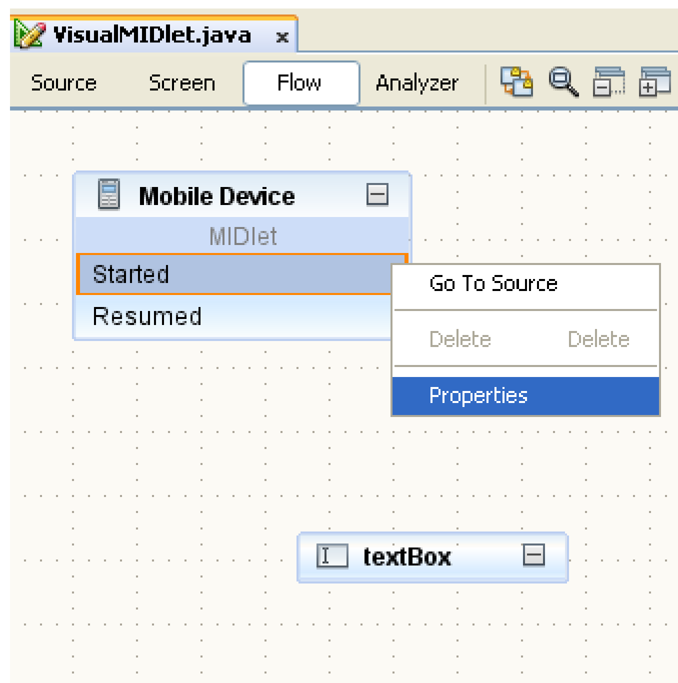


ايضا قم بالنقر المزدوج على القيمة <ull> وذلك لاضافة نص الرسالة "مرحبا بكم فى منتدى مصر التقنى " كما فى الصورة التالية:



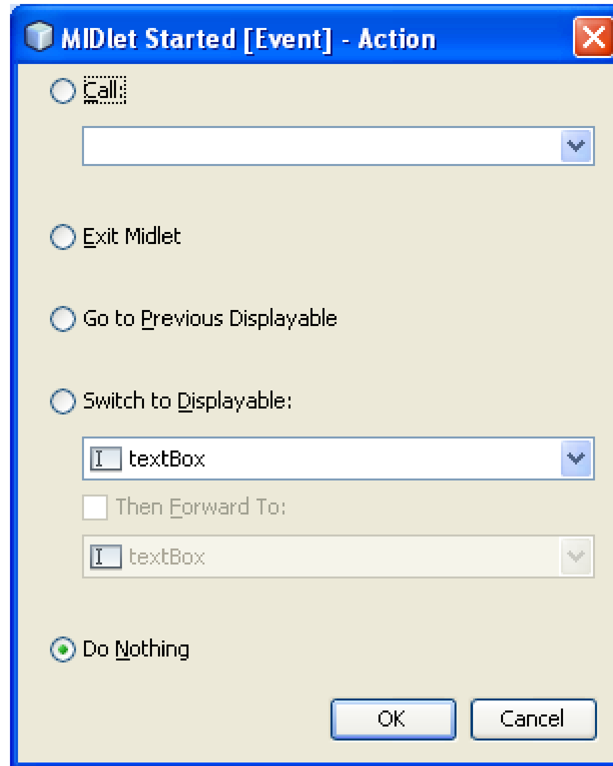
_3قم بربط المكون textbox بنقطة البداية للمورد او الملف MIDlet حتى يتم تشغيل التطبيق ويتم ذلك عن طريق الاوامر commandويتم ذلك عن طريقتين:

الطريقة الاولى : النقر على التويب Flow ثم تحديد الحدث Started للمورد MIDlet ثم النقر يمين ومن القائمة التى ستظهر أختر الامر propertites لتظهر لك النافذة MIDlet Started Event كما فى الصور التالية:





ويمكنك ملاحظة أن نافذة الخصائص تلك لا تحتوى إلا على خاصية واحده وهى... Action قم بالنقر على الزر الصغير الذى بجوارها ليتم لك فتح النافذة التالية:



النافذة السابقة تحتوى على ٥ خيارات للاحداث المتاحة و هى على النحو التالى:

1_الخيار..... call هذا الخيار يتم استخدامة فى حاله اذا اردنا الاتصال بأحد ملفات MIDlet الاخرى الموجودة بالتطبيق عند بداية التشغيل للتطبيق

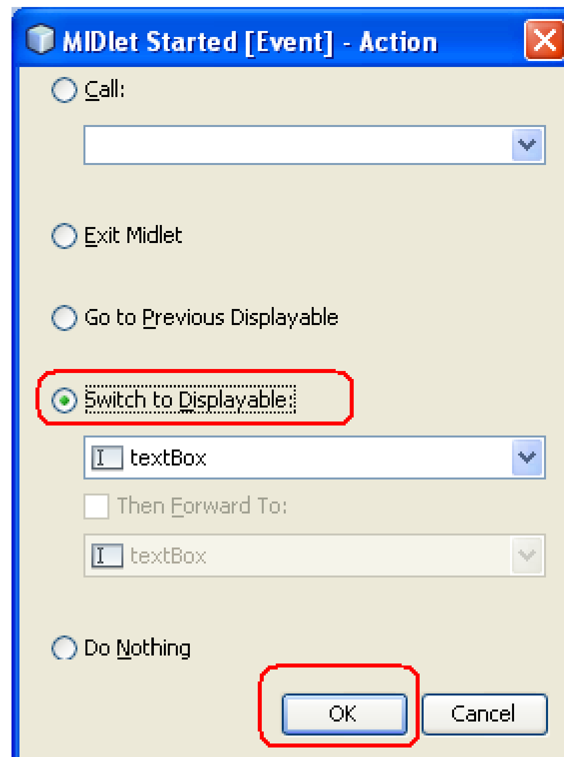
2_الخيار.... Eext MIDlet هذا الخيار يتم استخدامه اذا اردنا الخروج من MIDlet الحالى.

3_الخيار.... Go to previous Displayable هذا الخيار يتم استخدامه فى حاله وجود عده مكونات تابعة للفئة Displayableو تريد الرجوع للخلف الى المكون السابق

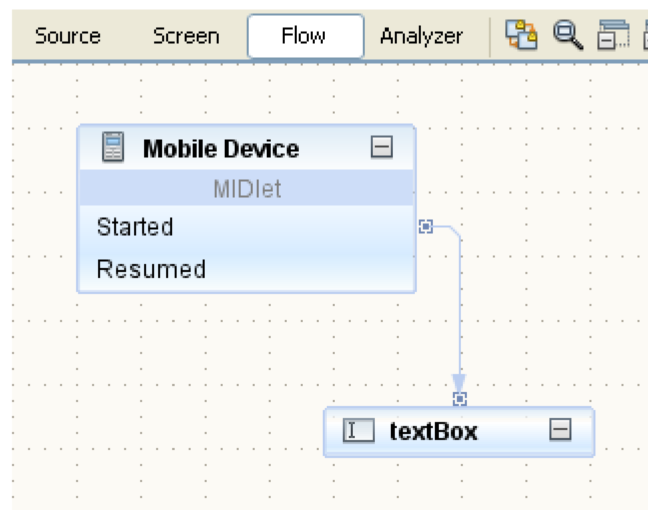
4_الخيار.... Switch to Displayable و هذا الخيار يتم استخدامه فى حاله اننا نريد تشغيل احد مكونات الفئة Displayableوتوجد قائمة بها جميع المكونات التى أضفتها لتطبيقك.

5_الخيار...... Do Nothing و هذا الخيار لعدم فعل اى حدث.

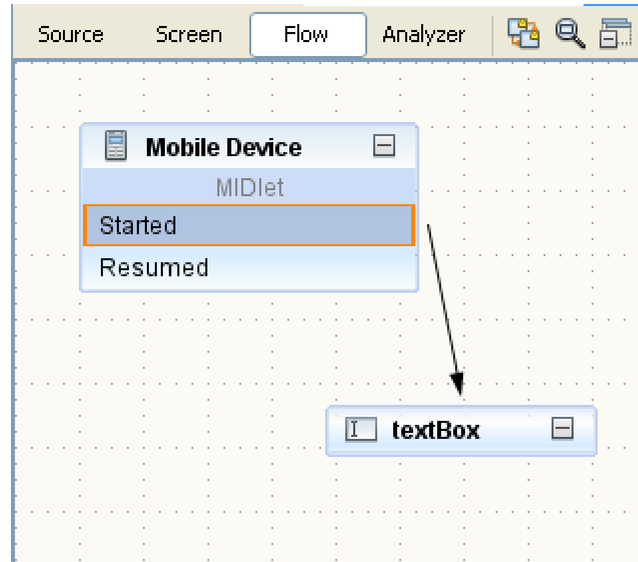
بعد ما تعرفنا على تلك النافذة نريد الان تشغيل المكون textbox بربطة بنقطة البداية Started للمورد MIDlet و ذلك من خلال الخيار الرابع Switch to Displayable و ستجد بجانبه قائمة بها المكونات التى تم إضافتها للتطبيق و فى مثالنا هذا هو مكون واحد فقط و هو textbox لذلك سنقوم بتحديد هذا الخيار ثم قم بالنقر على زر الامر(OK) كما فى الصورة التالية:



بعد ذلك ستجد مخطط التطبيق فى التويب Flow كما فى الصورة التالية:



الطريقة الثانية : قم بتحديد الحدث Started للمورد MIDlet ثم النقر بزر الماس شمال مع السحب فى وضعية النقر ليظهر لك سهم أسود أثناء السحب ثم قم بإقلاته على المكون المراد ربطة بنقطة البداية Started وفى مثالنا هذا هو المكون textboxكما فى الشكل التالى:

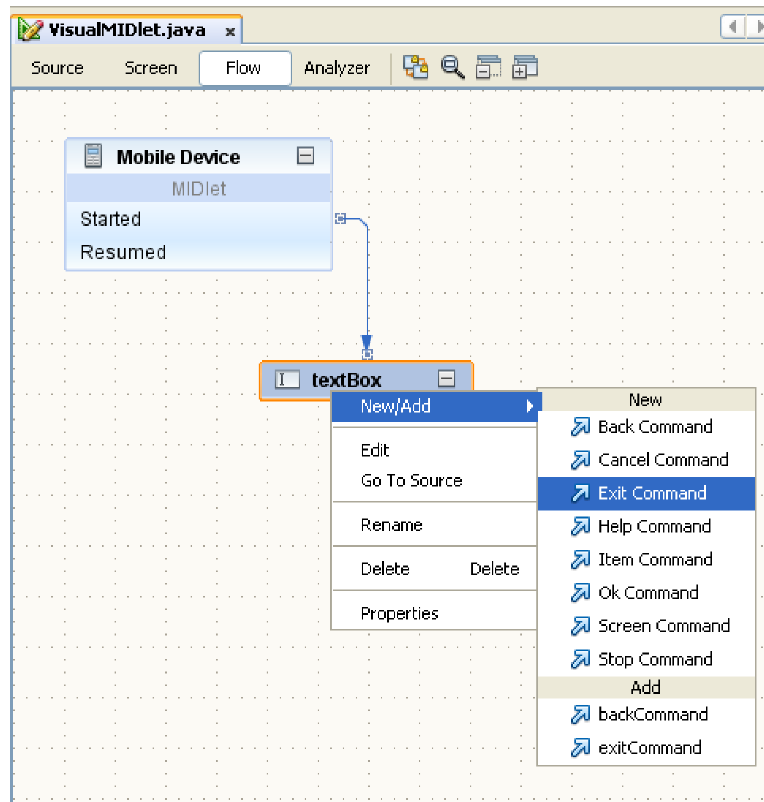


الان يمكنك تشغيل التطبيق لعرض رسالة الترحيب و يظهر كما فى الصورة التالية:



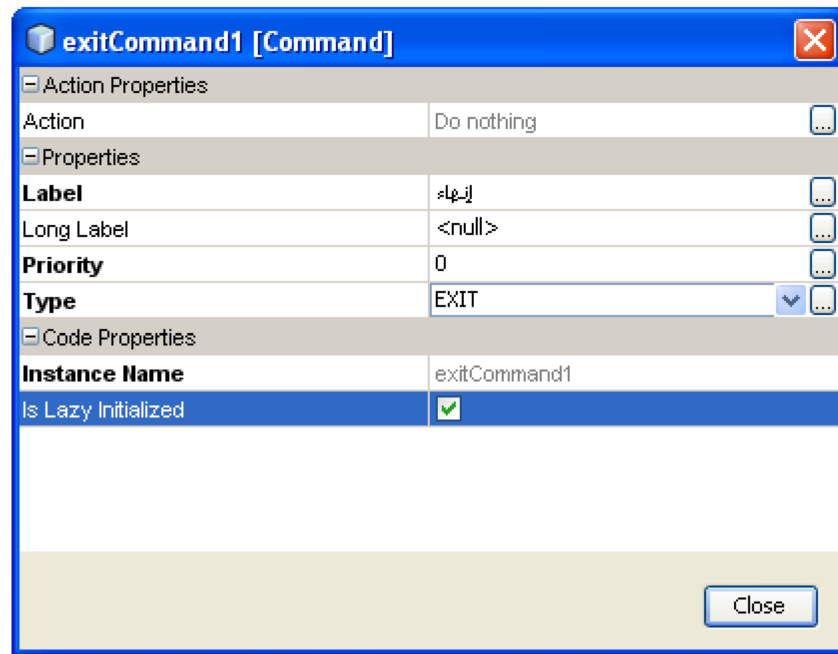
ويمكنك ملاحظة اننا قد قمنا بتشغيل التطبيق على محالكى اخر غير الذى تم استخدامه فى الدرس السابق ....الان تعالوا نقوم بإضافة بعض التنسيقات للتطبيق ...فمثلا نقوم بوضع أمر للخروج من التطبيق وذلك على النحو التالى:

_قم بتحديد المكون textbox من خلال التويب Flow ثم النقر يمين ومن القائمة التى ستظهر أختر الامر New/Add ثم أختر الامر Eixt command كما فى الصورة التالية:

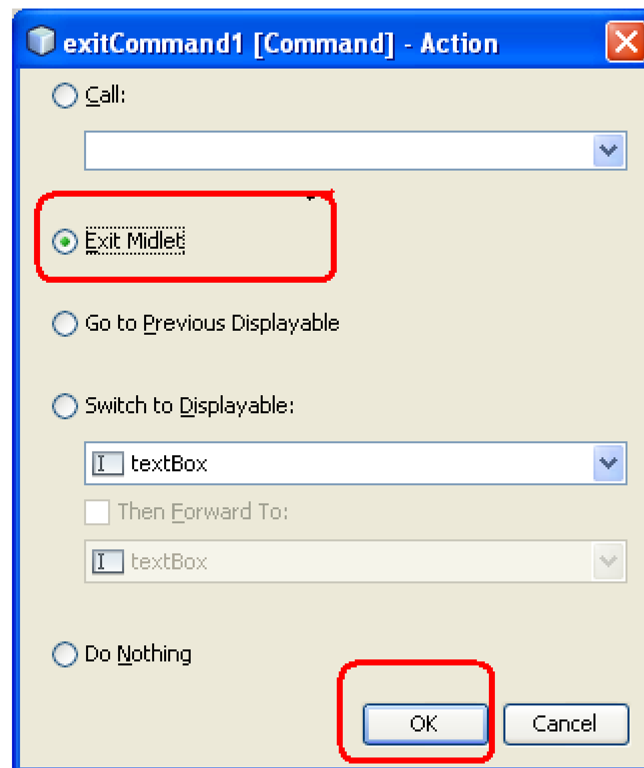


ستجد انه قد تم إضافة command للمكون textbox قم بتحديده ثم النقر يمين ومن القائمة التى ستظهر اختر الامر propertitesلتظهر لك نافذة الخصائص كما فى الصور التالية:

بعد النقر على زر (ok) ستجد فى التويب Flow التطبيق كما فى الشكل التالى:

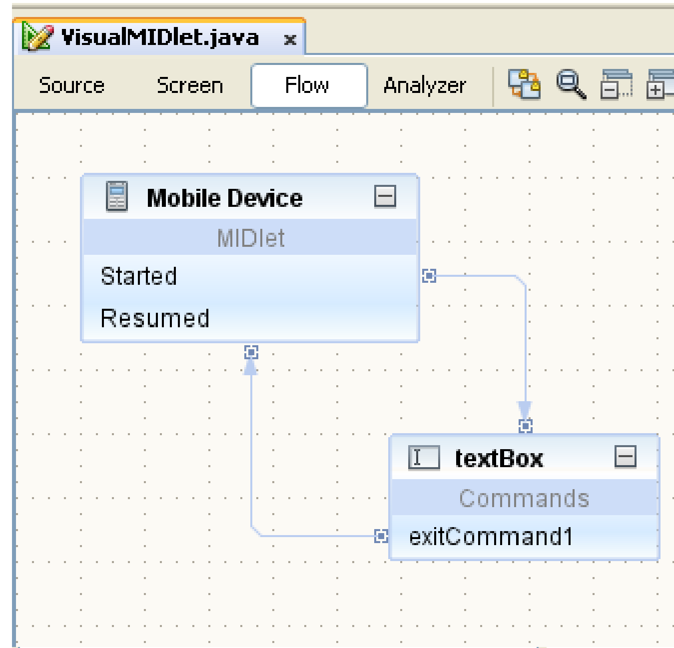


وكما هو ملاحظ فى نافذة الخصائص ....كتبنا كلمة (إنهاء)عند الخاصية label وذلك لوضع تسميه للامر تظهر فى الجوال ...أما الخاصية Action قم بالنقر على الزر الذى بجوارها لتحديد خيار الخروج من MIDlet كما فى الصورة التالية:



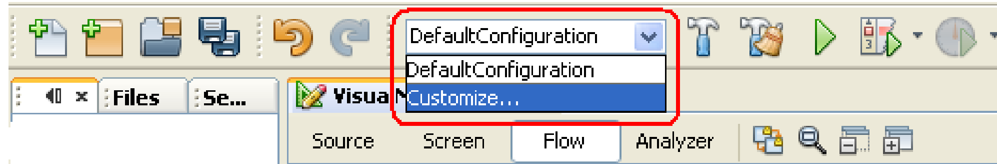
بعد النقر على زر (ok) ستجد فى التويب Flow التطبيق كما فى الشكل التالى:

ايضا يمنك ملاحظة اننا قد استخدمنا محاكى أخر غير المستخدم بالدرس السابق و غير المستخدم فى المثال السابق وهذا المحاكى الحالى يمثل أجهزة الجوال التى تعمل بخاصية اللمس وتستطيع أن تختار محاكى Device مختلف عما قد حددته فى بداية إنشاء المشروع عن طريق تحديد الامر Customize من قائمة المهام فى NetBeans كما فى الصورة التالية:

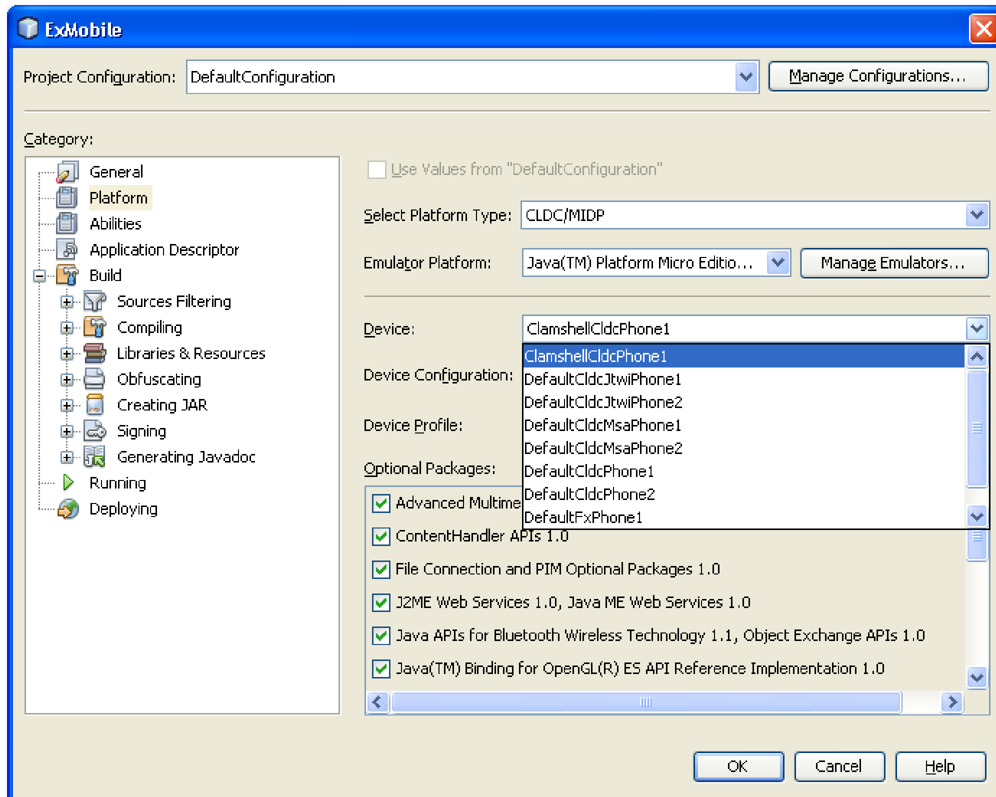


الان قم بتشغيل التطبيق ليظهر لك كما فى الشكل التالى:



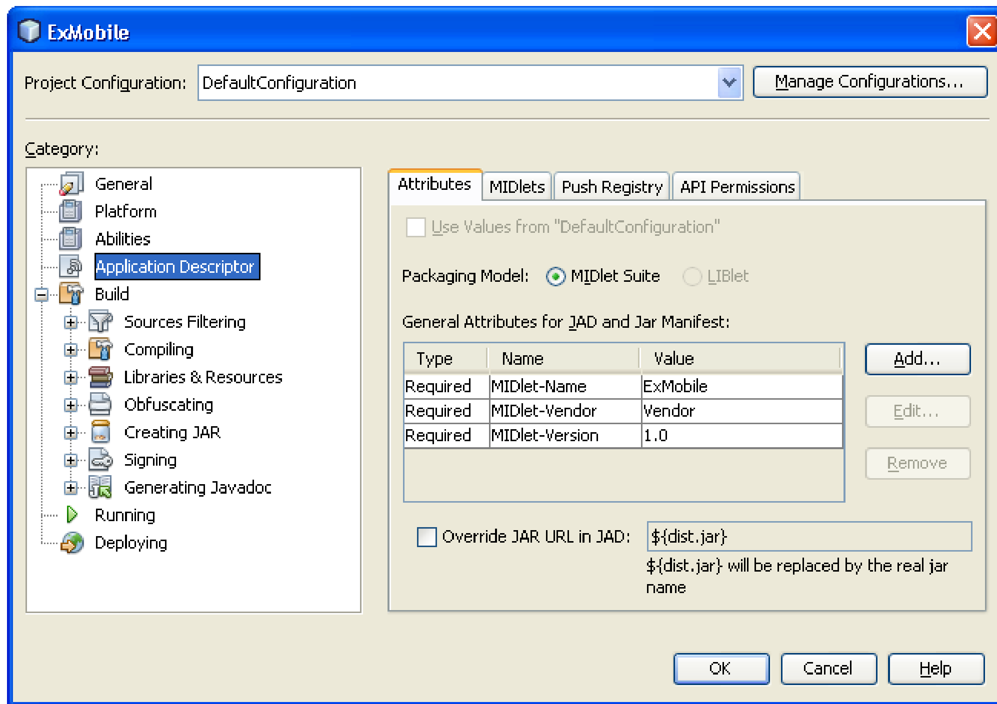


بعد ذلك سيظهر لك ٢ مربع حوارى الاول Add Configuration وهذا ليس لنا به علاقة الان لذلك قم بالنقر على زر الامر Cancel و المربع الحوارى الاخر كما فى الصورة التالية:



وكما هو ظاهر إعلاه يمكنك تحديد المحاكى الذى يناسبك من خلال القائمة Device و يجب ملاحظة أن كل محاكى يختلف عن الاخر و كلما كان المحاكى أحدث كلما كانت له مميزات كثيرة

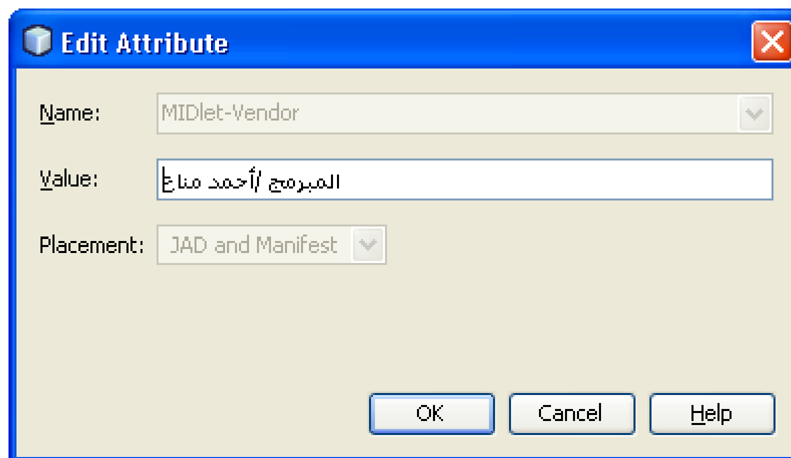
_تعالوا الان نضيف بعض الخصائص الاخرى للتطبيق.... فمن نفس المربع الحوارى السابق نستطيع إضافة رقم إصدار التطبيق و معلومات عن مبرمج التطبيق وايضا اضافة ايقونة للتطبيق كل ذلك من خلال المربع الحوارى السابق التويب Application Descriptionليظهر لك المربع الحوارى كما فى الصورة التالية:



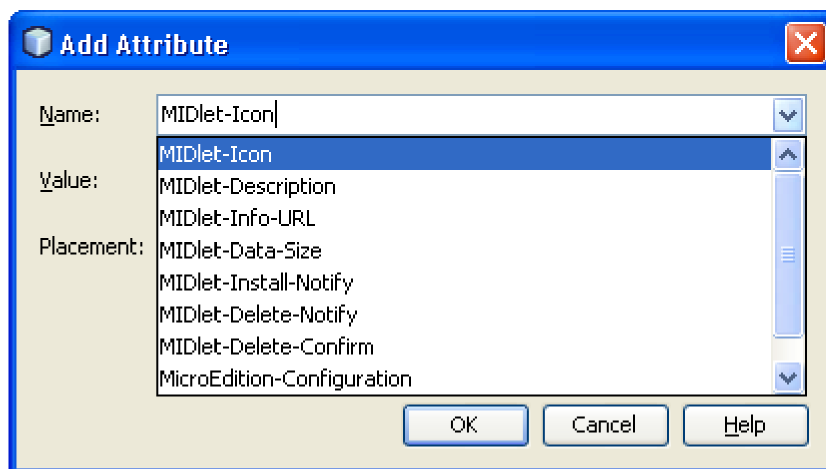
_كما هو ظاهر أعلاه ستجد جدول The General Attributes table به بعض المعلومات الافتراضة وهى على النحو التالى:

MIDlet-Name وهذا الخيار لوضع اسم للتطبيق
MIDlet-Vendor وهذا الخيار لوضع معلومات عن مبرمج التطبيق
MIDlet-version. وهذا الخيار لوضع رقم إصدار للتطبيق

وجميع تلك الخصائص اساسية لايمكنك حذفها ومع ذلك يمكنك تعديل القيم فقط بما يناسبك ......فقط قم بتحديد الخاصية المراد تعديلها و ستجد زر الامر (Edit) الموجود على يمين تلك الخصائص أصبح نشط لتقوم بالنقر عليه ليفتح لك مربع حوارى أخر تستطيع من خلاله تعديل البيانات لتلك الخصائص .....فأنا مثلا سأقوم بتعديل الخاصية-MIDlet Vendorلوضع اسم المبرمج لذلك سأقوم بتحديد تلك الخاصية ثم النقر على زر (Edit) ليتم فتح المربع الحوارى التالى:



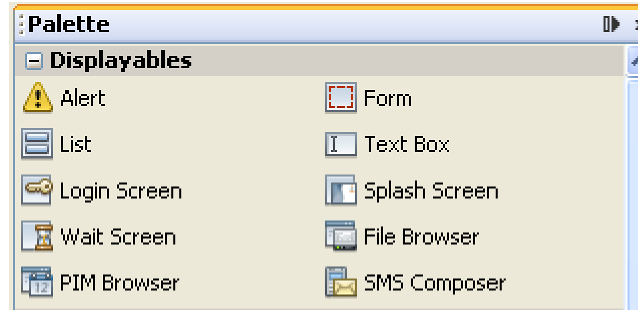
_كما يمكنك إضافة بعض الخصائص الاخر وذلك عن طريق زر الامر Add ليفتح لك مربع حوارى يحتوى على قائمة الخصائص المتاحة الاخرى كما فى الشكل التالى:



فمثلا أريد إضافة ايقونة للتطبيق لذلك سأقوم أولا بتحديد تلك الايقونة و يجب أن تكون بأمتداد BMP أو PNG و أن يكون حجمها مناسب لا تزيد عن ٦٤ بيكسل ثم وضع تلك الصورة فى مجلد البرنامج src سأختار من قائمة الخصائص الخاصية MIDlet-Iconثم أكتب أسم و أمتداد الصورة فى مربع النص value كما فى الشكل التالى:

الفصل الثانى : التعامل مع المكونات التابعة للتويب Displayable الجزء الاول!

خوانى الكرام أهلا وسهلا بكم فى منتدى مصر التقنى ....اليوم إن شاء الله تعالى سنتعلم سويا كيفية التعامل مع المكونات التابعة للتصنيف Displayable تلك المكونات يتم وضعها على الشاشة بدون أى حاوى لها و هى المكونات التالية:



1_المكونAlert

2_المكونForm

3_المكونList

4_المكونTextBox

والمكونات الاربعة أعلاه تنتمى للحزمةjavax.microedition.lcdui

5_المكونLogin Screen

6_المكونSplash Screen

7_المكونWait Screen

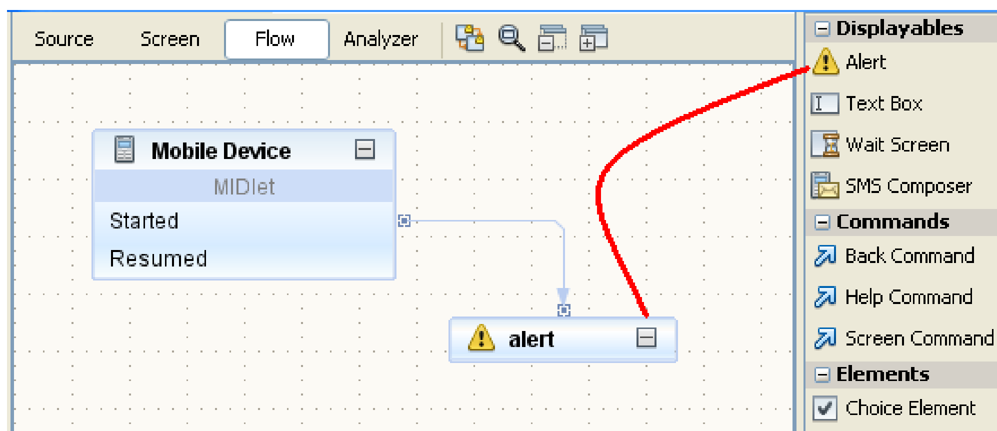
8_المكونFile Browser

9_المكونPIM Browser

10_المكونSMS Composer

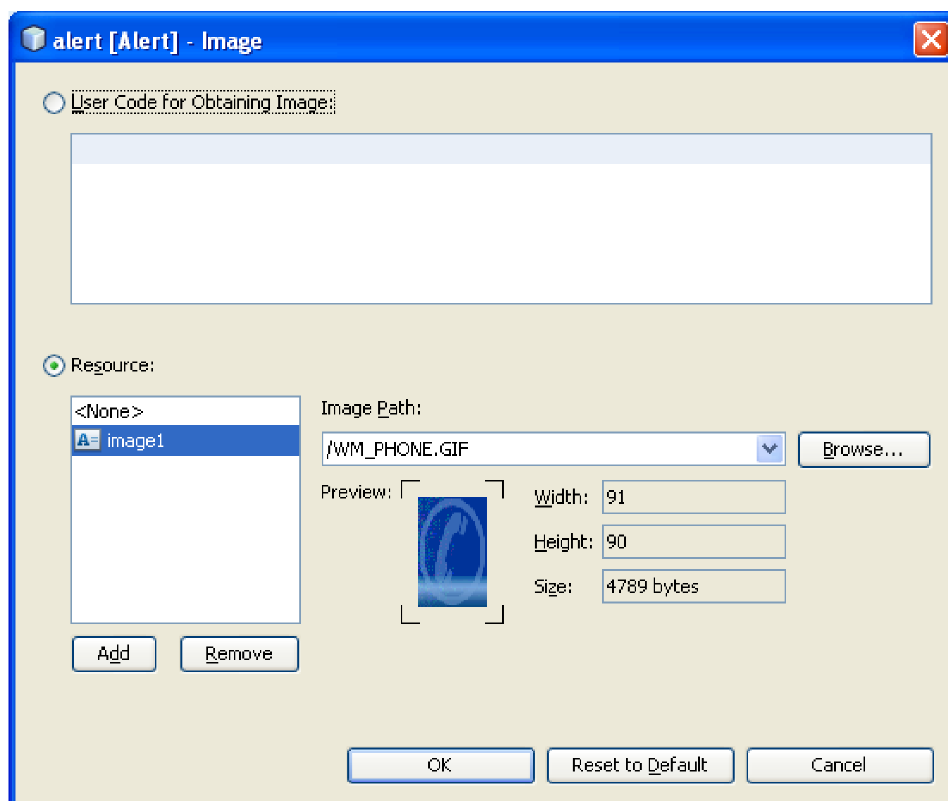
و المكونات الستة من ٥ الى ١٠ أعلاه تنتمى للحزمة org.netbeans.microedition.util وهى مكونات خاصة بالـ NetBeansو بالتالى ربما لن تجدها أذا كنت تستخدم محرر أخر ....تعالوا أن نتعرف على تلك المكونات

1_المكون.... Alert



هذا المكون يستخدم لعرض رسائل تنبيهيه أو تحذيرية أو معلومات وتنتظر لفترة معينة قبل الانتقال الى مكون أخر ...و تستطيع استخدام النصوص بهذه الرسالة أو استخدام صوره معينه و أهم الخصائص لهذا المكون ما يلى:

_الخاصية... Image و التى تتيح لك وضع صورة لرسالة التنبيه مع أو بدلا من النص ويمكنك الوصول لتلك الخاصية عن طريق نافذة الخصائص للمكون Alert ثم تحديد الخاصية Image ثم النقر على الزر الصغير الذى بجوارها لتظهر لك النافذة التالية و التى تستطيع إدراج صورة للمكون: Alert

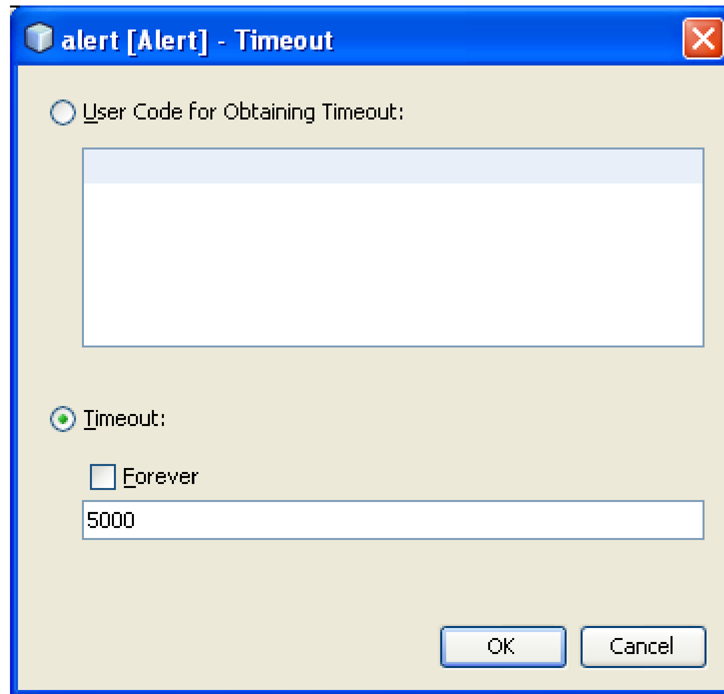


_الخاصية.... String و التى تتيح إدراج نص معين ليظهر فى المكونAlert

_الخاصية... Ticker و التى تتيح إنشاء شريط متحرك أسفل الرسالة.

_الخاصية... Timeout و التى تتيح وضع مقدار الزمن التى تستغرقه الرسالة فى ظهورها على الشاشة و تأخذ القيمة الافتراضية FOREVER و التى تعنى تبقى فى الظهور الى أن يتم النقر على حدث معين ....لكنك بالطبع تستطيع تغيرها

ووضع مده معينه و ذلك بالنقر على الخاصية ليتم فتح المربع التالى ...كل ما عليك فعله هو أزالة الخيار الموجود بجوار FOREVERليتم تنشيط مربع الزمن وضع فيه المده التى ترغب فيها و ليكن ٥ ثوانى لذلك ستكتب ٥٠٠٠ كما فى الصورة التالية:



2_المكون.....Form وهو عبارة عن نافذه لاحتواء المكونات التى لا يمكن استخدامها إلا فى وجودها مثل text fields و.... choice groups الخ ذلك من العناصر التى تطلب وجود نافذه ...أما اهم خصائص هذا المكون الخاصية Title التى تتيح وضع اسم للنافذه

3_المكون..... List وهى عبارة عن قائمة تحتوى على مجموعة من الخيارات ليختار من بينها المستخدم و لكى تستطيع إضافة عناصر تلك القائمة قم بتحديد المكون ثم النقر يمين ومن القائمة التى ستظهر أختر الامر Go to Source ليتم فتح نافذة الشفر و ستجد الكود التالى:

```
public List getList () {
if (list == null) {
list = new List ("list", Choice.MULTIPLE);
list.setCommandListener (this);
list.setFitPolicy (Choice.TEXT_WRAP_OFF);
}
return list;
}
```

سنقوم بإضافة العناصر بإستخدام الطريقة Append و ذلك بإضافة الكود التالى الى السابق وعلى النحو التالى:
الرمز:

```
public List getList () {
if (list == null) {

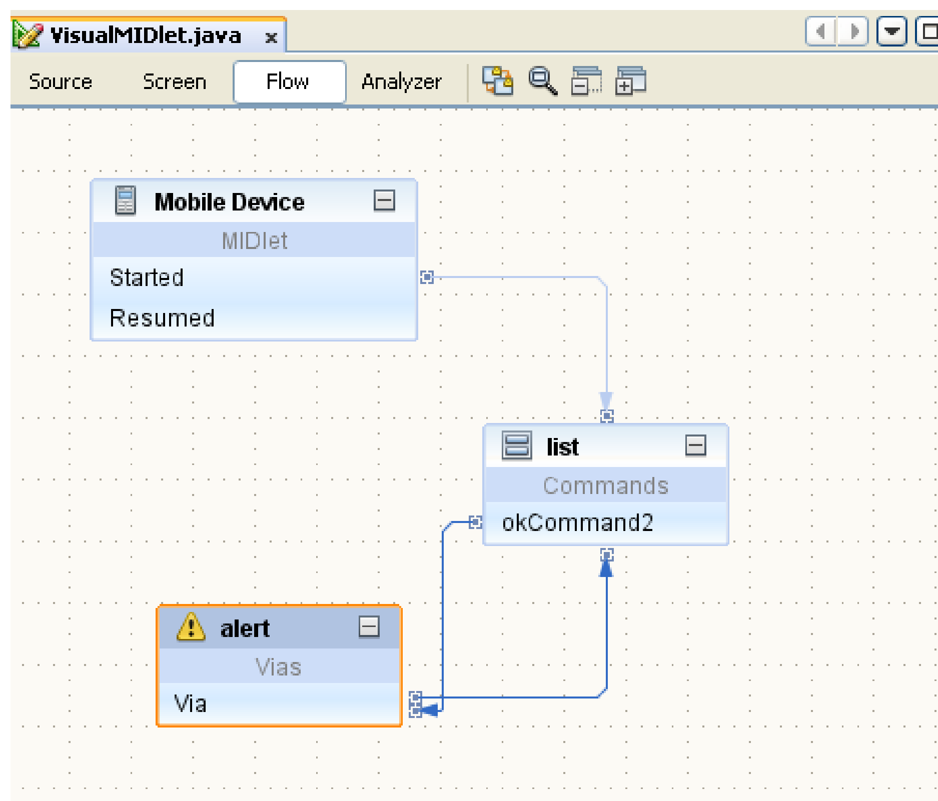
list = new List ("list", Choice.IMPLICIT);
list.setCommandListener (this);
list.setFitPolicy (Choice.TEXT_WRAP_OFF);
list.append("العنصر الاول", null);
```

```
list.append("العنصر الثانى", null);
list.append("العنصر الثالث", null);
list.append("العنصر الرابع", null);
list.append("العنصر الخامس", null);
}
return list;
}
```

و النتيجة كما فى الصورة التالية:



_كيفية تحديد عنصر من القائمة .... يمكنك من تحيد العنصر المختار من قبل المستخدم عن طريق الاسلوب getSelectedIndex() ولتوضيح الفكرة سنقوم بادراج المكون Alert مع المكون List مع وضع commandOK و تغير اسمه الى (موافق) كما فى الشكل التالى:



الان المطلوب هو عند تحديد عنصر من القائمة ثم النقر على موافق تظهر رسالة Alert بأسم العنصر المختار ....لذلك سنقوم بحديد المكون Alert ثم النقر يمين ليفتح لك نافذة الشفر لتجد الكود التالى بها:

الرمز:

```
public Alert getAlert () {
if (alert == null) {

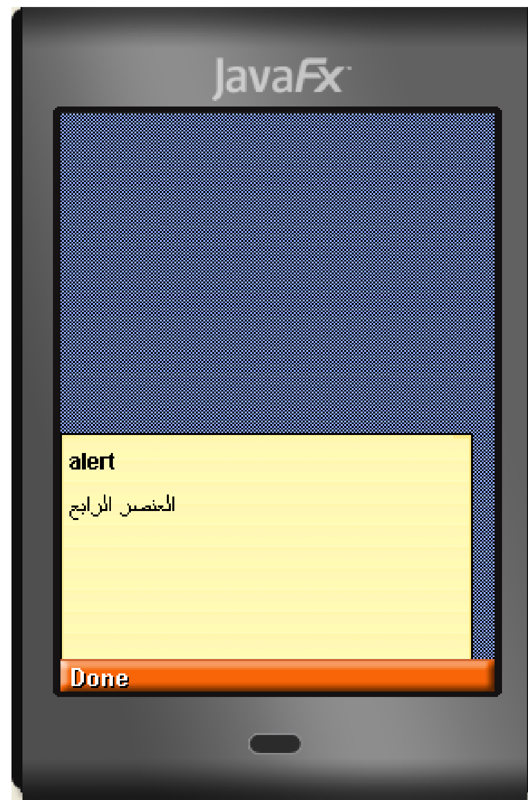
alert = new Alert ("alert");
alert.setTimeout (Alert.FOREVER);
}
return alert;
}
```

من جانبنا سنقوم بإضافة الكود التالى اليه:

```
int no=list.getSelectedIndex();
alert.setString(list.getString(no));
```

وكما هو ظاهر فى الكود أعلاه قمنا بالاعلان عن المتغير no ليقوم بحفظ الرقم المفهرس لكل عنصر ....و فى السطر الثانى النص الخاص بالرسالة alert قمنا بالحصول على اسم العنصر من خلال الرقم المفهرس بعد الحصول اسم العنصر المقابل لهذا للرقم الفهرس باستخدام الطريقة getString





_ومن أهم خصائص المكون list الخاصية Type و التى تشمل ثلاث خيارات ....وهى الخيار IMPLICIT و التى تقوم بعرض العناصر كما فى الامثلة السابقة ...و الخيار EXCLUSIVE و التى تقوم بعرض العناصر على خانات أختيار كما فى الشكل التالى :



أما الخيار الاخير هو MULTIPLE و الذى يقوم بعرض العناصر على هيئة مربعات اختيار متعدد كما فى الشكل التالى:



4_المكون.... TextBox هو عبارة عن شاشة تسمح للمستخدام بعرض وتحرير نص معين ويتميز بأنه يسمح بالاسطر المتعددة ....و أهم خاصية لهذا المكون هى الخاصية Text والتى من خلالها تستطيع تحرير النص .....و

ايضا الخاصية maximum size وهو الحد الاقصى لعدد الاحرف التى يمكن تخزينها فيه...

أما أهم الطرق أو الاساليب لهذا المكون هى الطريقة getString و التى يمكنك استخدامها للحصول على النص الموجود فى المكون و ايضا الطريقة setString و التى تستطيع من خلالها وضع نص معين اثناء التشغيل ...ايضا الطريقة size و التى تقوم بجلب عدد الاحرف الموجودة فى مربع النص

أما باقى العناصر التى تنتمى الى التويب Displayable سيتم شرحها فى درس أخر إن شاء الله تعالى فالى اللقاء

الفصل الثالث : التعامل مع المكونات التابعة للتويب الجزء الثانىDisplayable!

اهلا بكم إخوانى الكرام ...فى الجزء الثانى من درس كيفية التعامل مع المكونات التابعة للتصنيف Displayable ففى الجزء الاول أوضحنا أن هناك مكونات تابعة للحزمة javax.microedition.lcdui فى هذا الدرس إن شاء الله تعالى سنتعلم كيفية التعامل مع المكونات التابعة للحزمة org.netbeans.microedition.lcdui وهى على النحو التالى:

1_المكون....LoginScreen هذا المكون عبارة عن شاشة تسجيل دخول للتطبيق و التى تتطلب أسم المستخدم و كلمة مرور و هذا المكون مفيد للتطبيقات البنكية لما يتطلبه سرية المعلومات.

ومن أهم الخصائص التى يتميز بها هذا المكون ما يلى:

loginScreen [LoginScreen]

Properties	
Background Color	[0,153,153]
Foreground Color	[204,0,51]
Full Screen	☐
Login Button Text	دخول
Login Title	مرحبا بكم فى منتدى مصر التقنى
Password Label	كلمة المرور
Ticker	<None>
Title	شاشة تسجيل الدخول
Use Login Button	☑
Username Label	أسم المستخدم
Code Properties	
Instance Name	loginScreen
Is Lazy Initialized	☑
Login Properties	
Password	<null>
Username	<null>

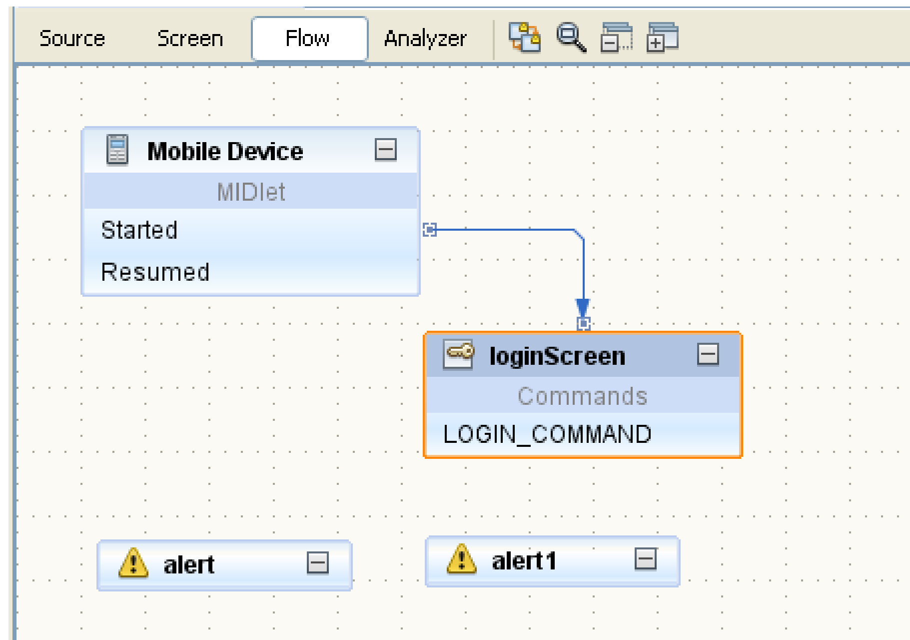
Close

_الخاصية.... Background Color لتحديد لون خلفية شاشة تسجيل الدخول
_الخاصية..... Foreground Color لتحديد لون الخط المستخدم فى شاشة الدخول
_الخاصية.... Use Login Button لعمل زر الدخول تحت حقل اسم المستخدم و حقل كلمة المرور
_الخاصية .... Login Button Text لكتابة نص على زر الدخول
_الخاصية............. Login Title وهى لوضع عنوان أو نص أعلى حقل أسم المستخدم وحقل كلمة المرور

تعالوا الان إخوانى الكرام نأخذ مثال بسيط على كيفية أستخدام هذا المكون ....سنقوم بعمل تطبيق يطلب من المستخدم أدخال أسما و كلمة المرور و إذا كانت البيانات التى أدخلها المستخدم صحيحة يتم اظهال رسالة بانه تم الدخول و العكس و ذلك على النحو التالى:

_قم بإضافة المكون LoginScreen وقم بتنسيق الخصائص اعلاه ....ثم قم بربط المكون بالحدث Started للمورد MIDlet

_قم بإضافة عدد ٢ مكون... Alert ومع تنسيق خصائصهم ....فالمكون الاول أجل الخاصية String مفداها) عفوا يوجد خطأ فى أسم المستخدم و كلمة المرور.... (و المكون الثانى) لقد تم الدخول بنجاح.....(على أن يكون شاشة التطبيق بهذا الشكل:



_قم بالذهاب الى صفحة كتابة الشفرة ثم حدد الطريقة public void commandAction (Command command, Displayable displayable) ...لتجد بها الكود التالى:

```
if (displayable == loginScreen) {
if (command == LoginScreen.LOGIN_COMMAND) {
```

سنقوم بتعديل الكود السابق بإضافة بعض الاكواد الية ليصبح كما يلى:

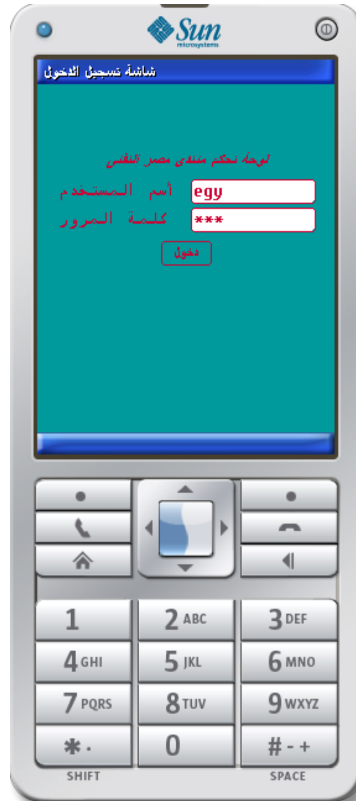
```
public void commandAction (Command command, Displayable displayable) {

if (displayable == loginScreen) {
if (command == LoginScreen.LOGIN_COMMAND) {

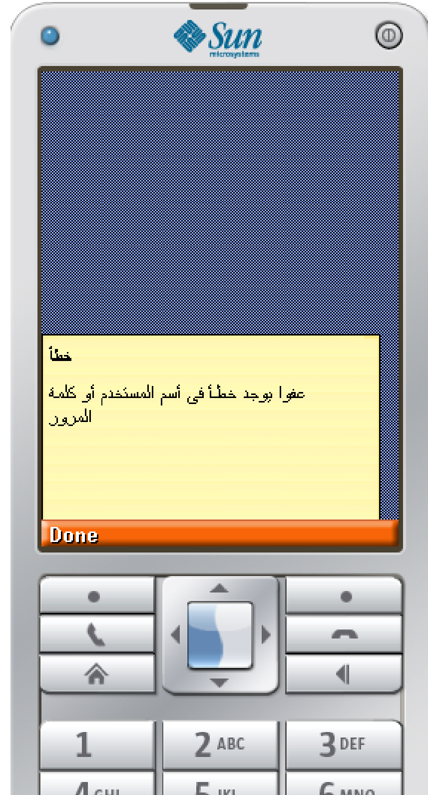
 String getuser=loginScreen.getUsername();
 String getpass=loginScreen.getPassword();

if (getuser.equals("egy") && getpass.equals("123")) {
switchDisplayable (getAlert (), getLoginScreen ());
} else {
switchDisplayable (getAlert1 (), getLoginScreen ());
}  }} }
```

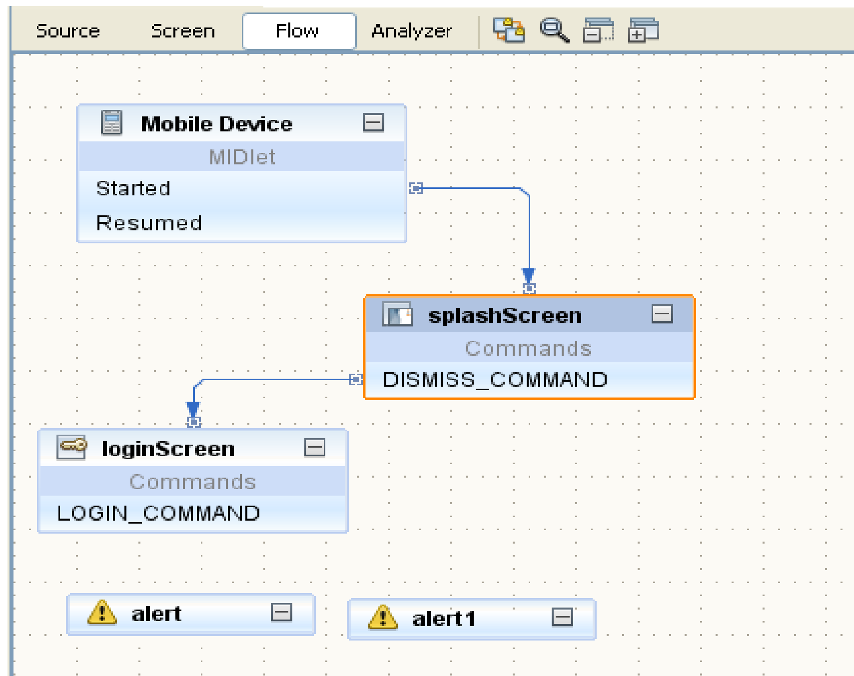
فى السطر الثالث قمن بالاعلان عن المتغير getuser و ذلك لحفظ اسم المستخدم المدخلة من المستخدم وذلك عن طريق الاسلوب.... loginScreen.getUsername اما فى السطر الرابع قمنا بالاعلان عن المتغير getpass لحفظ كلمة المرور من المستخدم بأستخدام الطريقة..... getPassword بعد ذلك أستخدمنا جملة شرطية للتأكد من القيم المدخلة من المستخدم و مقارنتها بأسم المستخدم (egy) و كلمة المرور (١٢٣) فإن كانت القيم المدخلة صحيحة يتم اظهار رسالة يخبرنا بذلك و العكس صحيح .....و النتيجة كما فى الشكل التالى:


شاشة تسجيل الدخول
لوحة تحكم منتدى مصر التقنى
أسم المستخدم
egy
كلمة المرور
***
دخول
SHIFT
SPACE


دخول ناجح
أهلا وسهلا بكم فى لوحة تحكم المنتدى
Done



2 المكون.....**SplashScreen** وهو عبارة عن شاشة افتتاحية تظهر لمدة معينة اثناء تحميل التطبيق ....ومن اهم خصائص هذا المكون الخاصية Image التى تتيح وضع صورة للشاشة .... الخاصية Text و التى تتيح لك كتابة نص فى الشاشة .... الخاصية Timeout و التى تحدد المدة الزمنية لابقاء الشاشة ظاهرة ثم تختفى ....فى مثالنا السابق سنقوم بادراج المكون SplashScreen ثم قم بربط المكون بالحدث Started للمورد.... MIDlet بعد ذلك ربطه بالمكون loginScreenكما فى الشكل التالى:



و النتيجة كما فى الشكل التالى:



3_المكون.....Wait Screen وهو عبارة عن شاشة انتظار حتى تنفيذ أمر ما ....ومن اهم خصائص هذا المكون الخاصية Image التى تتيح وضع صورة للشاشة ....الخاصية Text و التى تتيح لك كتابة نص فى الشاشة ....وسيتم التعرف على كيفية استخدامة فى الامثلة التالية

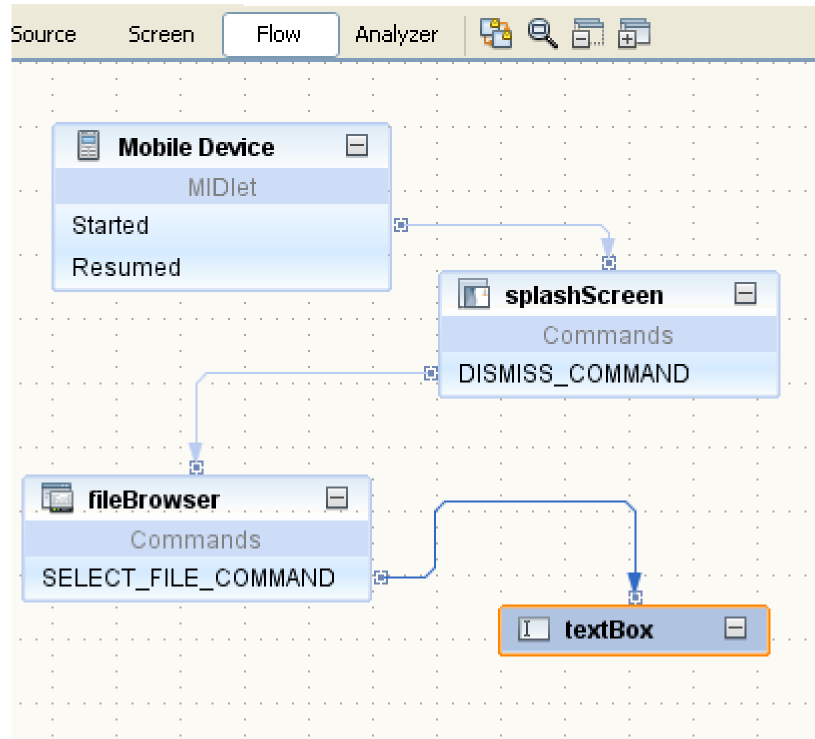
4_المكون... File Browser وهو عبارة عن مكون مخصص الذي يوفر واجهة مستخدم للعمل مع نظام ملفات الجهاز.حيث يستخدم لتصفح محتويات ذاكرة التخزين و الجوال ......اهم ما يميز هذا المكون الطريقة التابعة له getSelectedFileوالتى من خلالها تستطيع تحديد او الحصول على الملف الذى تبحث عنه داخل الذاكرة...

مثال :لقراءة ملف من ذاكرة الهاتف....

_قم بوضع المكون SplashScreen و ربطة بالحدث Started للمورد MIDlet

_قم بوضع المكون.... File Browser وربطة بالحدث DISMISS_COMMAND الخاص بالمكون SplashScreen

_قم بإضافة المكون TextBox و ربطة بالحدث SELECT_FILE_COMMAND الخاص بالمكون File Browserكما فى الصورة التالية:



_سنقوم بإنشاء الطريقة التالية:

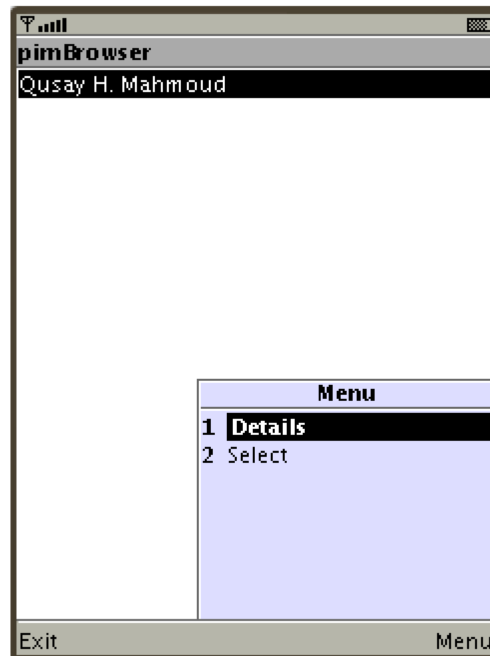
```
private void readFile() {

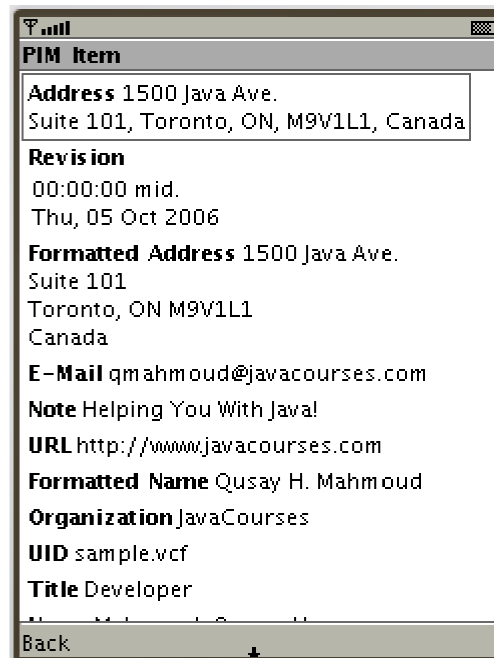
        try {
          FileConnection textFile = fileBrowser.getSelectedFile();
          getTextBox().setString("");
          InputStream fis = textFile.openInputStream();
          byte[] b = new byte[1024];
          int length = fis.read(b, 0, 1024);
          fis.close();
          if (length > 0) {
              textBox.setString(new String(b, 0, length));
          }
      } catch (IOException ex) {
      }

  }
```

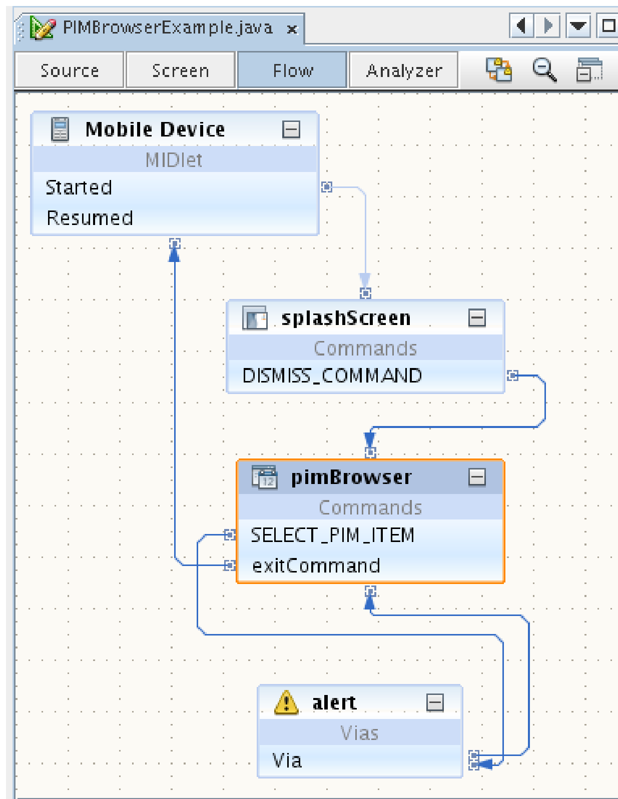
_نقوم بتحديد الطريقة commandAction وبعد سطر
الامر FileBrowser.SELECT_FILE_COMMAND نقوم بادراج الطريقة التى قمنا بإنشائها;readFile()

4_ المكون.... **PIM Browser** وهو عبارة عن مستعرض يمكنك للوصول الى مهام الجوال بسهولة مثل دفتر العناوين و الاسماء المخزنة و التقويم و قائمة المهام و غيرها ....واهم ما يميز هذا المكون ان يحتوى على عدد ٢ زر امر.... commanaالاول details و الذى يقوم بعرض كافة البيانات المتعلقة بالاختيار المحدد ...فمثلا لو تم اختيار احد الاسماء تستطيع معرفة كافة التفاصيل لهذا الاسم





الزر الثانى SELECT و الذى من خلاله تستطيع تنفيذ اى امر ...فمثلا نريد اظهار رسالة باسم المحدد ....نقوم بعمل البرنامج كما فى الشكل التالى:



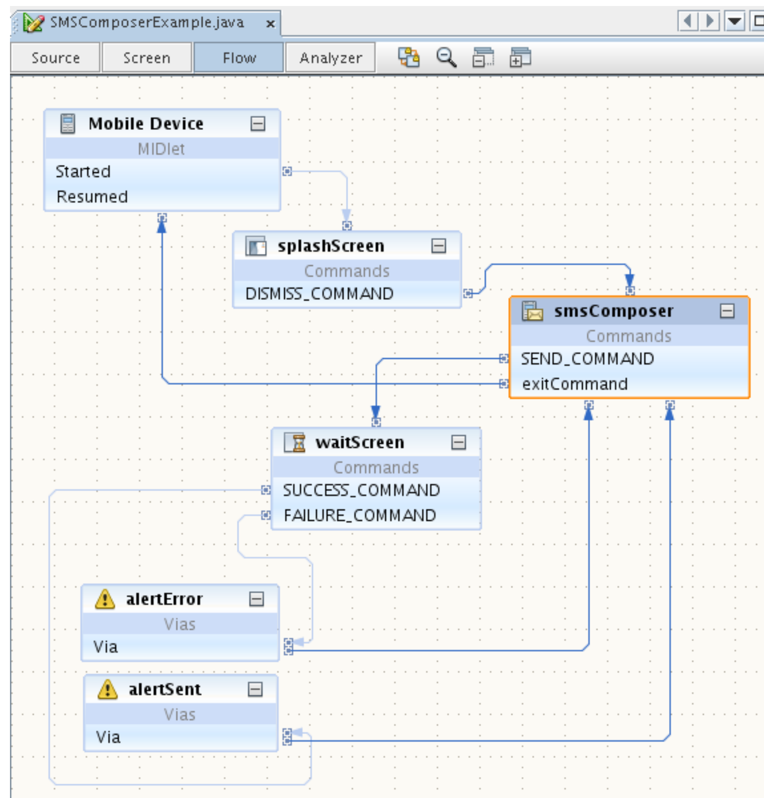
5 المكون.... **SMS Composer** يستخدم هذا المكون لارسال الرسائل النصية القصيرة.... sms ومن اهم الخصائص المميزة لهذا المكون الخاصية Message Label و هى عنوان مربع الرسالة ...الخاصية Phone Number Labelوهى عنوان مربع رقم المرسل الية ... الخاصية Message و هى الخاصية التى تتيح لنا كتابة الرسالة المراد ارسالها .... الخاصية Phone Number و التى تتيح لنا كتابة رقم جوال المرسل الية

sms Composer
Phone Number:
Message:
Exit
Send

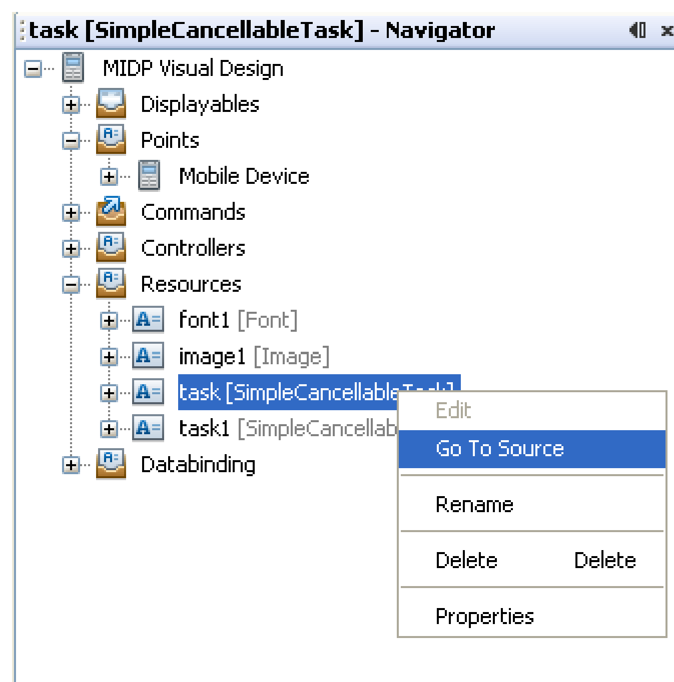
مثال ....نقوم بادراج المكونات التالية

Splash Screen
SMS Composer
Wait Screen
Alert (عدد ٢)

نقوم بربط المكونات السابقة كما فى الشكل التالى:



_من قائمة Navigator نحدد التصنيف Resources ثم نبحث عن task ثم النقر بزر الفارة اليمين ومن القائمة التى ستظهر أختر الامر go to source كما فى الصورة التالية:



لتفتح لك نافذة كتابة الشفرة فنكتب الامر التالى:

```
smsComposer.sendSMS();
```

طبعا سيتم كتابة الامر فى المنقطة التى يوجد بها التعليقwrite task-execution user code here

الفصل الرابع : التعامل مع المكونات التابعة للتويب Items

أهلا وسهلا بكم إخوانى الكرام ....فى الدرسين السابقين تعرفنا على المكونات التابعة للتصنيف Displayable اليوم أن شاء الله تعالى سنتعرف سويا على المكونات التابعة للتصنيف... Items وتلك المكونات لابد لها من حاوى بمعنى انك لا تستيطع وضعها على الشاشة بدون نافذة وبالتالى أستخدام تلك المكونات لابد أولا من أدراج Form ليتم وضع تلك المكونات عليه .....واهم تلك المكونات ما يلى:

1_المكون.... ChoiceGroup هو عبارة عن قائمة من العناصر يسمح للمستخدم للاختيار فيما بينهما

مثال _ : قم بإضافة المكون Form لمشروعك حتى يتسنى لنا إضافة المكون ChoiceGroup عليه
_من خلال التويب... Screen قم بسحب المكون ChoiceGroup ثم وضعه على النافذه Form
_قم بتحديد المكون ChoiceGroup ثم النقر يمين و من القائمة التى ستظهر أختر الامر Go to Source ليتم فتح نافذة الشفر
_سنقوم بكتابة الكود التالى فى المنطقة المكتوب فيها التعليق write post-init user code here و ذلك على النحو التالى:

```
choiceGroup.append("الفول", null);
choiceGroup.append("الطعمية", null);
choiceGroup.append("الفراخ", null);
choiceGroup.append("الكباب", null);
choiceGroup.append("المحشى", null);
choiceGroup.append("مكرونة", null);
```

الكود السابق هو لاضافة العناصر الى هذا المكون و ذلك من خلال الاسلوب append و النتيجة كما فى الشكل التالى:



كما يمكنك تحديد نوع القائمة تلك و ذلك من خلال الخاصية... Type ففى المثال السابق كانت نوع القائمة تساوى MULTIPLEو التى تجعل القائمة على هيئة مربعات أختيار متعددة كما فى الصوره أعلاه .....كما يمكنك جعل القائمة على هيئة خانات أختيار فردى بجعل الخاصية Type تساوى EXCLUSIVE كما فى الصوره التالية:

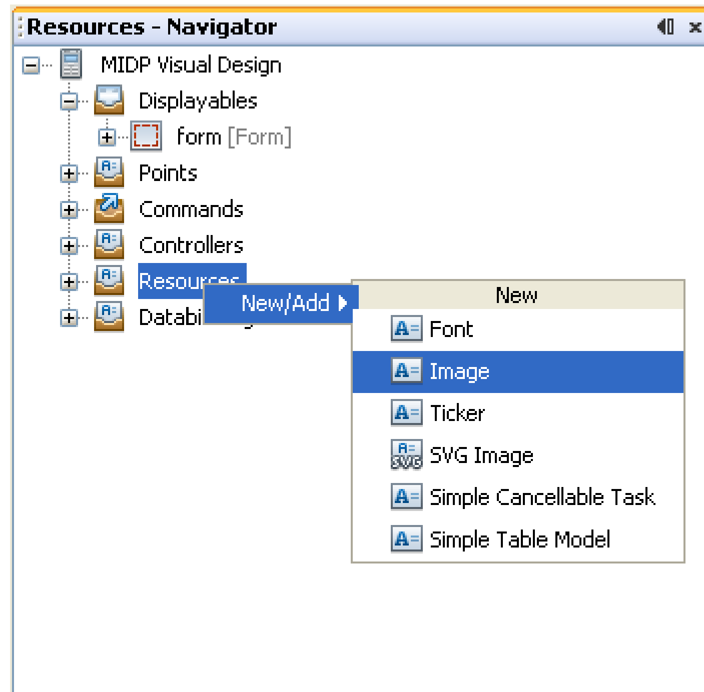


اما اذا أردت أن تجعل القائمة على هيئة مربع سرد combox أجعل الخاصية Type تساوى POPUP فتصبح القائمة كما فى الشكل التالى :

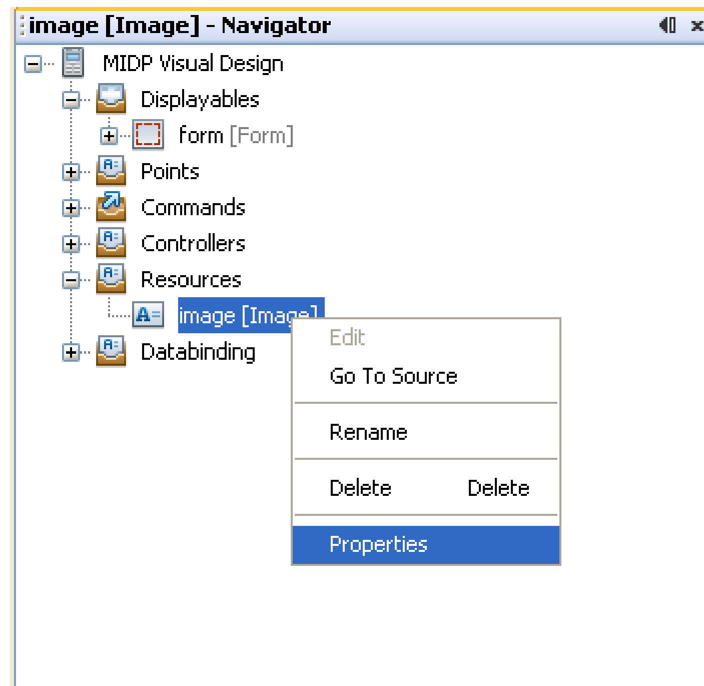


كما يمكنك وضع ايقونة أمام كل عنصر من عناصر تلك القائمة .... وذلك على النحو التالى:

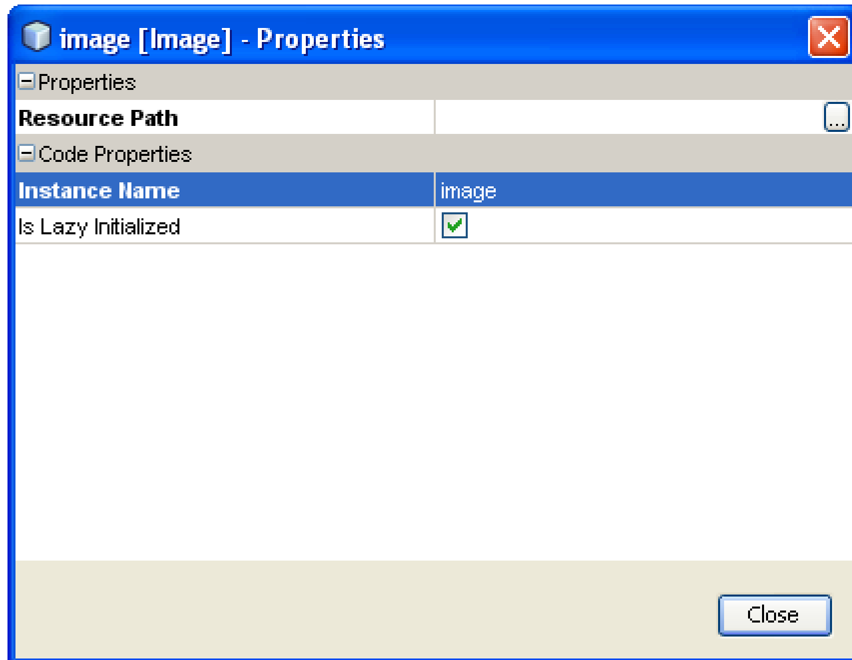
_ من قائمة Navigator نحدد التصنيف Resources ثم النقر يمين و من القائمة التى ستظهر اختر الآمر New/Add ثم أختر الامر Image كما فى الصورة التالية:



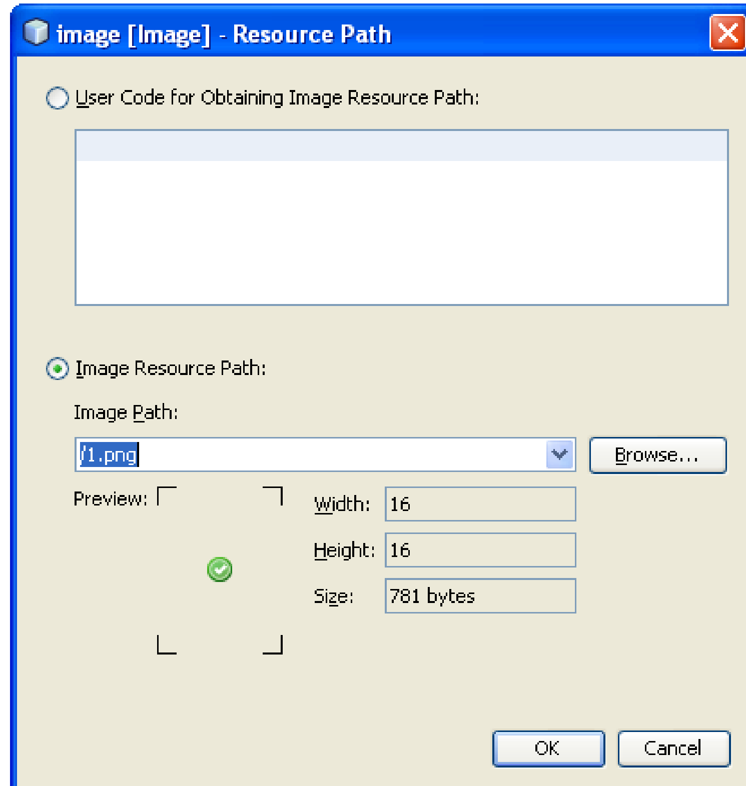
_ستجد بعد ذلك ايقونة Image قم بالنقر يمين عليها و من القائمة التى ستظهر أختر الامر properties كما فى الصورة التالية:



ليتم فتح لك نافذة الخصائص كما فى الشكل التالى:



حدد الخاصية Resource Path ثم انقر زر الامر الصغير الذى بجانبها ليتم فتح النافذة التالية:



من خلال المربع الحوارى السابق يمكنك اختيار صورة مناسبة من خلال زر الامر (أستعراض) ثم أنقر (موافق)

_نذهب الان الى صفحة كتابة الشفرة ....ونحدد الكود الذى قد كتبناه لاضافة العناصر الى القائمة ثم نقوم بتعديله و ذلك من خلال مسع الامر null ووضع مكانه الامر getImage ليصبح كما فى الشكل التالى:

```
choiceGroup.append("الفول ", getImage ());
choiceGroup.append("الطعمية", getImage ());
choiceGroup.append("الفراخ", getImage ());
choiceGroup.append("الكباب", getImage ());
choiceGroup.append("المحشى", getImage ());
choiceGroup.append("امكرونة", getImage ());
```

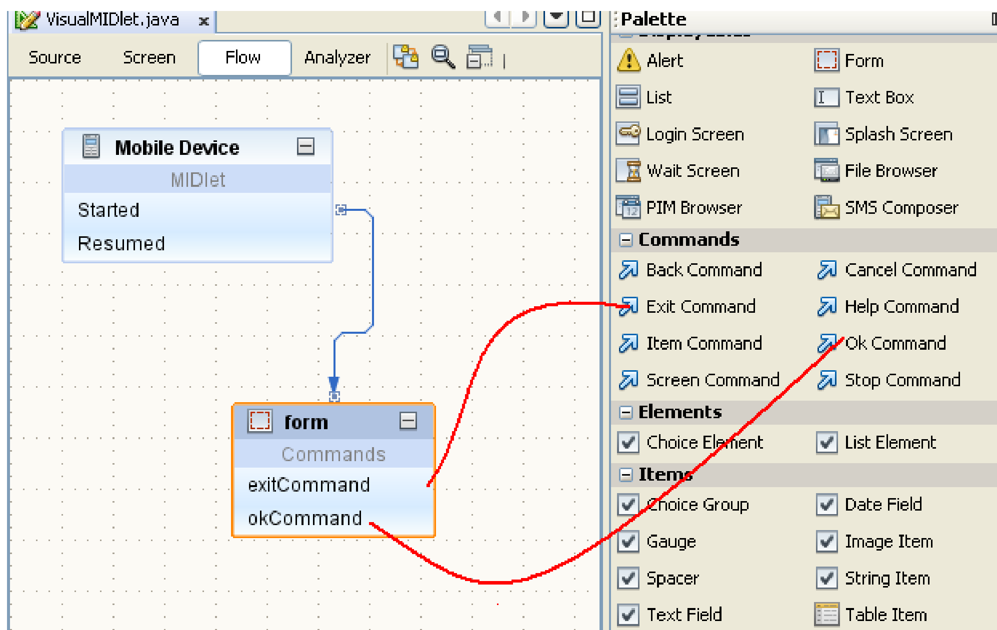
و النتيجة كما فى الاشكال التالية:

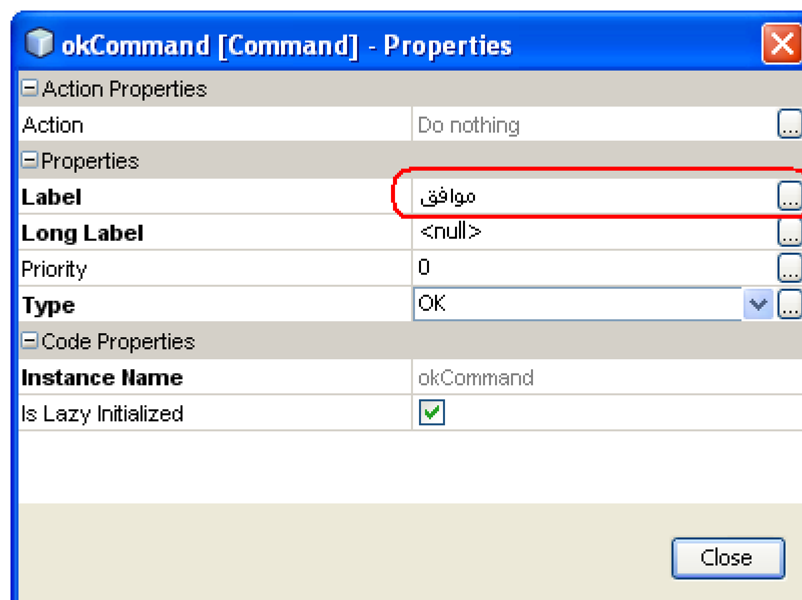




_كيفية أختيار عنصر من القائمة....

_لكى نستطيع فعل ذلك قم بإضافة عدد.... 2 command الاول قم بتنسيقه و ذلك بجعل الخاصية label تساوى خروجو تحديد الخاصية type تساوى... exit أما الثانى أجعل الخاصية label تساوى موافق و الخاصية type تساوى... ok وطبعا يتم اضافتهم من خلال التويب flow كما فى الشكل التالى:





_نضيف ايضا المكون Alert و ذلك لاظهار رسالة بالعنصر المختار من قبل المستخدم
_نقوم بتحديد المكون Alert ثم النقر يمين و من القائمة التى ستظهر أختر الامر GO TO Source ليتم فتح لك نافذة الكود و بداخل الطريقة Alert نكتب الكود التالى:
الرمز:

```
String selectedvalue=choiceGroup.getString(choiceGroup.getSelectedIndex());
alert.setString(": "+selectedvalue+".");
```

و أعتقد أن الكود السابق قد تم شرحة عندما تكلمنا عن المكون list فى الدروس السابقة.

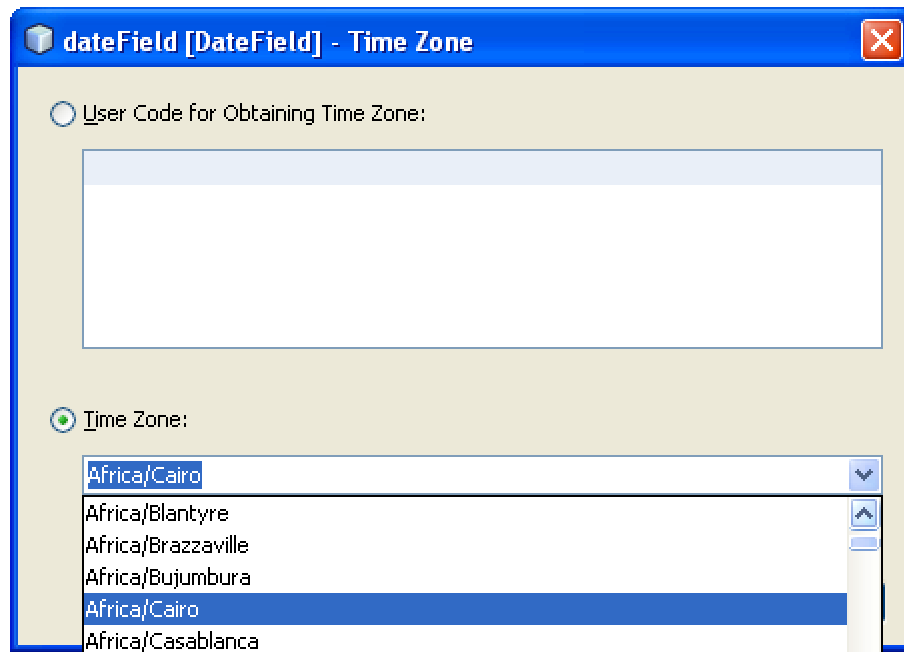
2_المكون.... DateField يستخدم هذا المكون لعرض بيانات التاريخ و الوقت (التقويم)و على النافذة..



ويعتمد حسابات هذا التقويم في على اللغة الافتراضية للجهاز ومنطقة زمنية محددة...ومن خصائص هذا المكون الخاصية Input Modeو التى تشمل ثلاث خيارات:

... DATE _ و التى تقوم بعرض التاريخ فقط
... DATE_TIME _و التى تقوم بعرض التاريخ و الوقت معا
.... TIME _و التى تقوم بعرض الوقت فقط

و من اها الخصائص ايضا لهذا المكون الخاصية Time Zone و التى تقوم بعرض الوقت و التاريخ العالمى و الموحد فقط ما عليك فعله هو النقر على زر الصغير بجوار تلك الخاصية ليتم فتح النافذة التالية:...



ومن النافذة اعلاه حدد المنطقة المراد تشغيل الوقت لها.

3المكون.... Gauge هو عبارة عن رسم بيانى أو شريط تقدمى يقدم بعض المعلومات من قيمة عددية.. ...التي تقع بين الصفر وأقصى قيمة...و يمكنك استخدام هذا المكون فى تطبيقات المديدياكمؤشر لقيمة الصوت



من اهم الخصائص المميزة لهذا المكون الخاصية Gauge Value و التى تحدد القيمة الحالية للعنصر ....و الخاصية Maximum Valueو التى تحدد اقصى قيمة لهذا العنصر .... أما الخاصية Is Interactive تجعلك قادر

على تغير القيمة الحالية للعنصر أثناء التشغيل و يأخذ العنصر الشكل التالى:



ومن أهم الطرق التابعة لهذا المكون ما يلى:

.... getMaxValue _للحصول على اقصى قيمة للعنصر
.... getValue _للحصول على القيمة الحالية للعنصر

.... 4_ImageItem هو أحد العناصر الذى يمكنك وضع صوره به لعرضها على النافذة مثلا ....من أهم الخصائص المميزه له الخاصية Image التى من خلالها تستطيع وضع الصورة التى تريدها...

_5المكون.... StringItem هو احد العناصر الذى يستخدم لعرض نص معين ولا يمكن المستخدم من تعديله وهو شبيه للعنصر jlabel او بمعى اصح مربع تسمية .... من اهم الخصائص المميزة لهذا العنصر الخاصية Text و هو النص الذى سيتم عرضة .....و ايضا الخاصية Appearance و التى تحدد مظهر العنصر و تأخذ ثلاث قيم PLAIN و هو المظهر الافتراضى HYPERLINK _ و هو التى تعطى للمكون مظهر ارتباط تشعبى BUTTON _ و هى القيمة التى تعطى للمكون مظهر زر الامر كما فى الصورة التالية:

ومن أهم الطرق الخاصة بهذا المكون:...

_الطريقة...... getText للحصول على النص الموجود بهذا المكون

_الطريقة.... setText لوضع نص بالمكون

_الطريقة... setPreferredSize و التى تقوم بتحديد قيم الطول و العرض لهذا المكون

_6المكون..... TextField هو عبارة عن حقل نصى قابل للتحرير مثل مربع النص textbox الذى تعرفنا عليه فى بداية هذه الدروس و الفرق بينهما هو أن حقل النص TextField لا قبل كتابة أسطر متعدده على غرار مربع النص.... textboxوايضا حقل النص لابد أن يضع على نافذه و لايمكن استخدامه بدون نافذه ....كما أن حقل النص له حد اقصى من الحروف وهى يقدر بحوالى ٣٢ حرف غير ذلك سيكون حجمة أكبر من حجم شاشة الجوال....

من أهم الخصائص المميز لهذا المكون ما يلى :

_الخاصية..... Input Constraints وهى عبارة عن قيود أدخال مختلفة يمكنك استخدام أحدها و ذلك لاجبار المستخدام أدخال بيانات معينة و التى تشمل القيود التالية:

ANY
EMAILADDR
NUMERIC
PHONENUMBER
URL
DECIMAL

فمثلا اذا قمنا باختيار القيمة NUMERIC معنى ذلك سيتم اجبار المستخدم من أدخال ارقام فقط ...ولو استخدمنا PHONENUMBERيتم أجبار المستخدام أستخدام الفواصل أرقام وعلامات الترقيم بما يتناسب مع الاتفاقيات لرقم الهاتف قيد الاستعمال ....كما يمكنك تنسيق القيمة المدخلة من خلال عدة خيارات اخرى و تشمل:

PASSWORD
UNEDITABLE
SENSITIVE
NON_PREDICTIVE
INITIAL_CAPS_WORD
INITIAL_CAPS_SENTENCE

_الخاصية..... Input Modes وهى خاصية تيح لك تحديد وضع الادخال للعنصر عند التركيز عليه ....بمعنى عندما يتم تحديد العنصر و لغة الكتابة الافتراضية لجوالك هى الانجليزية يمكنك تحديد تلك الخاصية بحيث يتم تعديل وضع الكتابة من الانجليزية الى العربية مباشرة بمجرد تحديد هذا المكون بدلا من الذهاب إعدادات الجوال لتغير لغة الكتابة الى العربية ....أو يتم تحديد وضع الادخال على الارقام

الفصل الخامس : التعامل مع نظام إدارة السجلات RMS

أهلا وسهلا بكم إخوانى الكرام ....تعرفنا فى الدروس السابق على المكونات الرسومية التابعة للتويب Displayableو ايضا المكونات التابعة للتويب.... Items درس اليوم سنتعلم فيه سوياً كيفية التعامل مع Record Management System وقبل أن نبدأ تعالوا نتعرف على بعض المفاهيم:

أولا : مفهوم نظام أدراة السجلات (RMS)

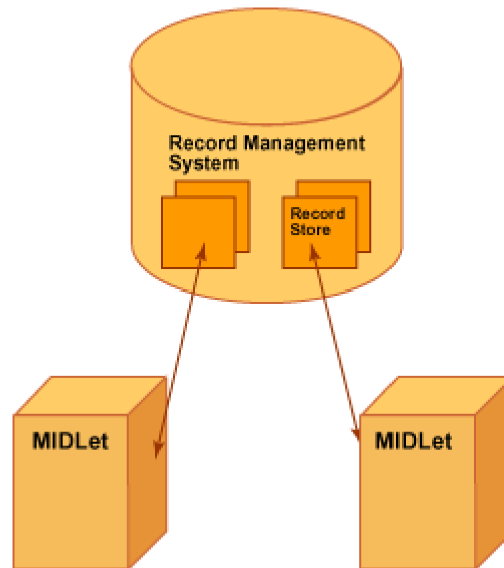
نظام إدارة السجلات RMS هو إختصار للجملة الانجليزية Record Management System وهو آلية أو قاعدة بيانات بسيطة المنحى للتخزين البيانات باستمرار واسترجاعها في وقت لاحق و التى يوفره المورد MIDlets أو معلومات الملف الشخصي الجهاز المحمول

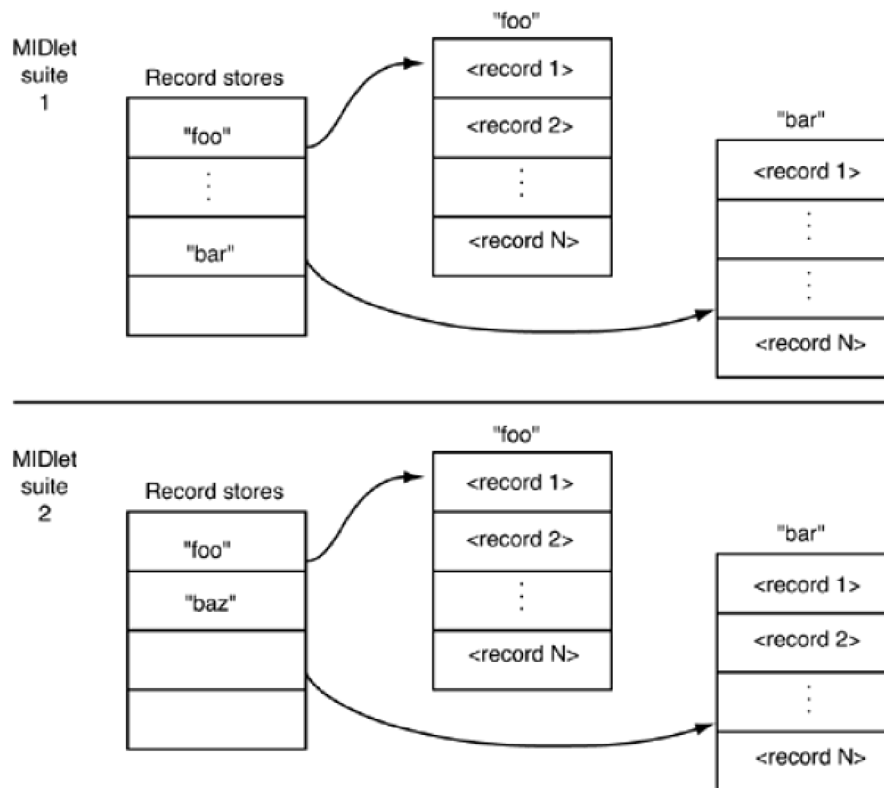
ثانيا : مفهوم السجلات Records

السجلات هى عبارة عن مصفوفة من البيانات من نوع bytes تمكن المطور من تخزين البيانات فيها عن طريق الكائن DataOutputStreamو الكائن DataInputStream و ايضا الكائن ByteArrayOutputStream ويتم تحديد السجلات بشكل فريد و إعطاء كل سجل مفتاح اساسى بحيث يبدأ برقم واحد للسجل الاول داخل ما يسمى "مخزن السجلاتRecordStore "

ثالثا : مفهوم مخزن السجلات RecordStore

هو عبارة عن مجموعة من السجلات المرتبطة بمنصة MIDlets و التى ستبقى ثابتة معه ...ويتم تسمية كل مخزن سجل RecordStoreبأسم فريد لتميز كل مخزن عن الاخر ويجب أن يكون الاسم من نوع String ولا يزيد عن ٣٢ حرف كما أن أسماء مخزن السجلات حساسة لحالة الاحرف فمثلا " EgyTech"ليست مثل"egytech "





1_إنشاء مخزن السجلات.... RecordStoreتوجد عدة طرق و اساليب لانشاء مخزن سجل RecordStore و ذلك على النحو التالى:

بداية يجب أستيراد المكتبة اللازمة rms.RecordStore فى اى من الطريق التى سيتم التعرف عليها و ذلك عن طريق الكود التالى:

```
import javax.microedition.rms.RecordStore;
```

الطريقة الاولى : الاعلان عن مخزن السجل RecordStore و إعطاء اسم مناسب له بأستخدام الكود التالى:

```
private RecordStore myegy = null;
```

كما ترى أخى الكريم فى الكود أعلاه قمنا بالاعلان عن مخزن السجل myegy وجعلنا القيمة له تساوىnull

الطريقة الثانية: فتح و انشاء مخزن السجل بدون أى شروط باستخدام الكود التالى:

```
RecordStore.openRecordStore("myegy",true);
```

الطريقة الثالثة: فتح و انشاء مخزن السجل بشرط جعل منصات MIDlet الاخرى قادرة للوصول اليه و السماح بالكتابة فيه و ذلك با ستخدام الكود التالى:

```
RecordStore.openRecordStore("myegy", true,2, true);
```

وكما ترون فى الكود أعلاه يتضمن ٤ عوامل:

.... myegy _وهو أسم مخزن السجل المراد إنشائه

... true _وهى لانشاء المخزن الذى قمنا بتحديد اسمه و يمكنك جعلها flase أذا كنت قد أنشأتها بالفعل

_الرقم ٢.... وهو متغير من نوع int للتعبير عن حالة الوصول هذا المخزن و تأخذ قيمتين
١ AUTHMODE_PRIVATE_و التى تعنى جعل المخزن myegy لا يمكننا الوصول اليه من خلال المنصات الاخرى .....٢ AUTHMODE_ANY_اى يمكن الوصول الية

.... True _متغير منطقى للتعبير عن أمكانية الكتبة فى هذا المخزن من خلال المنصات الاخرى

2_الاستثناءات الناتجة من فتح و إنشاء مخزن السجل RecordStore
...
عند استخدامك لاحد الطرق السابقة فى انشاء وفتح مخزن السجل RecordStore سينتج الاستثناءات التالية:

.... RecordStoreException _وهو خطأ يظهر فى حالة وجود مشكلة ما اثناء فتح وانشاء مخزن السجل

...RecordStoreNotFoundException _وهو خطأ يظهر فى حالة محاولة فتح مخزن السجل غير موجود ....كأستخدامك مثلا الطريقة الاولى لانشاء المخزن واستخدام الامر التالى لفتحه:

```
RecordStore.openRecordStore("myegy",false);
```

ففى الكود السابق نحاول فتح المخزن myegy وقد قمنا باستخدام الامر false أى أن المخزن قد انشأناه من قبل

..... RecordStoreFullException _هذا الخطأ يظهر فى حالة كون مخزن السجل ممتلىء

..... IllegalArgumentException_هذا الخطأ يظهر فى حالة كونك تستخدم اسم لمخزن السجل غير صالح للاستخدام.

3 _إغلاق مخزن سجل

```
myegy.closeRecordStore();
```

4_حذف مخزن سجل

```
RecordStore.deleteRecordStore("myegy");
```

5_الحصول على حجم مخزن السجل:

```
myegy.getSize();
```

و الكود السابق يقوم بإرجاع مقدار المساحة، بالبايت،

6 _إضافة سجل الى مخزن السجلات:.

يتم إضافة سجل الى مخزن السجلات باستخدام الكود التالى:

```
myegy.addRecord(byte[] data, int offset, int numBytes)
```

وكما ترون فى الكود أعلاه يطلب منك ثلاث عوامل لاضافة سجل جديد وهى على النحو التالى:

المعامل الاول .... data وهى البيانات التي سيتم تخزينها في هذا السجل ويجب أن تكون البيانات ذات طول
صفري أى مصفوفة من نوع byte..... قد تكون هذه المعلمة فارغة null اذا اردت إضافة سجل فارغ
المعامل الثانى... offset رقم المؤشر إلى المخزن المؤقت للبيانات من أول بايت ذات الصلة لهذا السجل
المعامل الثالث.... numBytes وهى عدد وحدات البايت المراد حجزها من المخزن myegy لهذا السجل

مثال:

```
byte[] name ="ahmed".getBytes();
 myegy.addRecord(name,0,name.length);
```

فى السطر الاول قمنا بعمل مصفوفه من نوع byte اسمها name ووضع فيها القيمة.... ahmed أما السطر الثانى فقمنا باضافة سجل جديد لمخزن السجل RecordStore باستخدام الاسلوبaddRecord
مع تحديد البيانات المراد تخزنيها بالسجل وهى فى مثالنا هذا المصفوفة name ثم وضعنارقم المؤشر لهذا السجل (٠)...وايضا عدد الواحدات لهذا السجل و ذلك باستخدام الاسلوبname.length

_قراءة سجل

```
byte[] getRecord(int recordId)
```

كما ترون فى الكود اعلاه تجد أن الاسلوب getRecord يقوم بإرجال قيمة السجل بناء على رقم المعرف الخاص به record Idفقلد أخبرنا بداية الدرس أنه يتم حديد السجلات بشكل فريد و إعطاء كل سجل مفتاح اساسىid بحيث يبدأ برقم واحد للسجل الاول ...أذن السجل الذى قمت باضافة فى المثال السابق المفتاح الاساسى له هو رقم ١ ...و بالتالى فإن القراءة ستكون كما يلى:

```
byte[] data = myegy.getRecord(1);
```

_تحديث سجل... اذا اردت تحديث سجل موجود من قبل ببعض البيانات نستخدم الكود التالى:

```
 byte[] name ="syed".getBytes();
 myegy.setRecord(1,name,0,name.length);
```

طبعا رقم ١ هذا هو الرقم المعرف للسجل الذى قمنا بإضافته فى بداية المثال

_**حذف سجل**.... اذا اردت حذف سجل أستخدم الكود التالى:

```
myegy.deleteRecord(1);
```

_**عدد السجلات فى مخزن السجل**....... يمكنك تحديد عدد السجلات فى مخزن السجل الحالى باستخدام الكود التالى:

```
myegy.getNumRecords();
```

_**الحصول على حجم السجل بالبايت**

```
myegy.getRecordSize(1);
```

مثال أكثر توضيحا :

بعد ما تعرفنا فى الجزء الاول من هذا الدرس عن كيفية التعامل مع نظام إدارة السجلات سنتعلم سويا انشاء تطبيق عملى وليكن مخزن كتب....

الخطوة الاولى : قم بإضافة المكون form وذلك من خلال التويب flow وسحب المكون من القائمة Displayable

الخطوة الثانية : قم بإضافة المكون TextField ويتم وضعة على النافذة form

الخطوة الثالثة : نقوم بإضافة عدد ٨ اوامر Command من نوع itemcommand ويتم تنسيقهم على النحو التالى:

Label إنهاء

Instance Name exit

Label فتح المخزن

Instance Name open

Label إضافة

Instance Name add

Label حذف

Instance Name delet

Label عدد السجلات

Instance Name no

Label عرض

Instance Name view

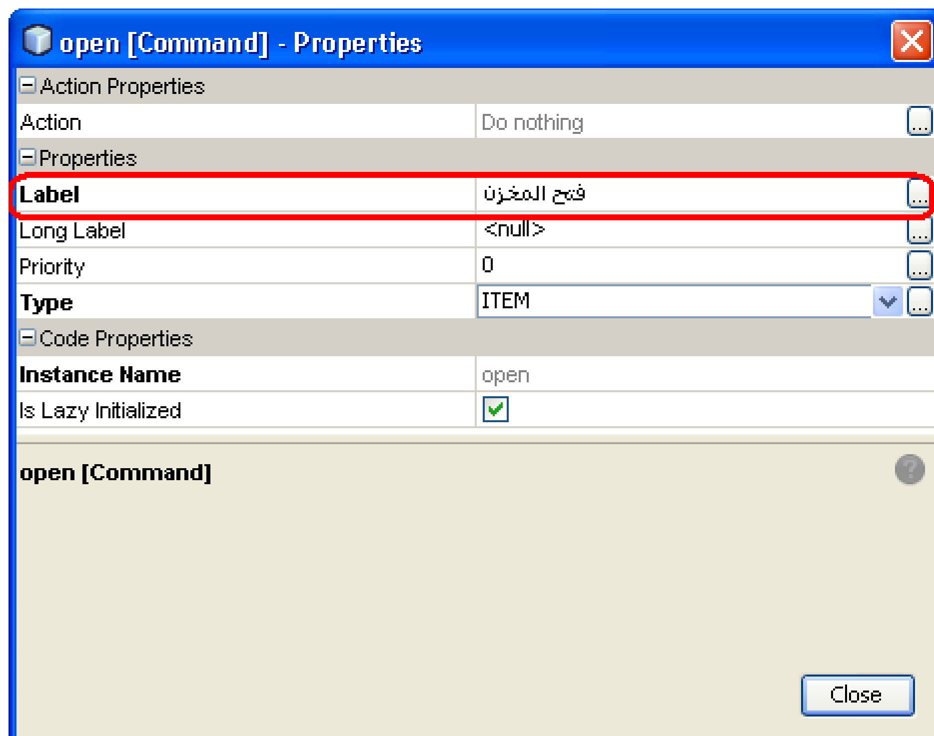
Label إغلاق

Instance Name close

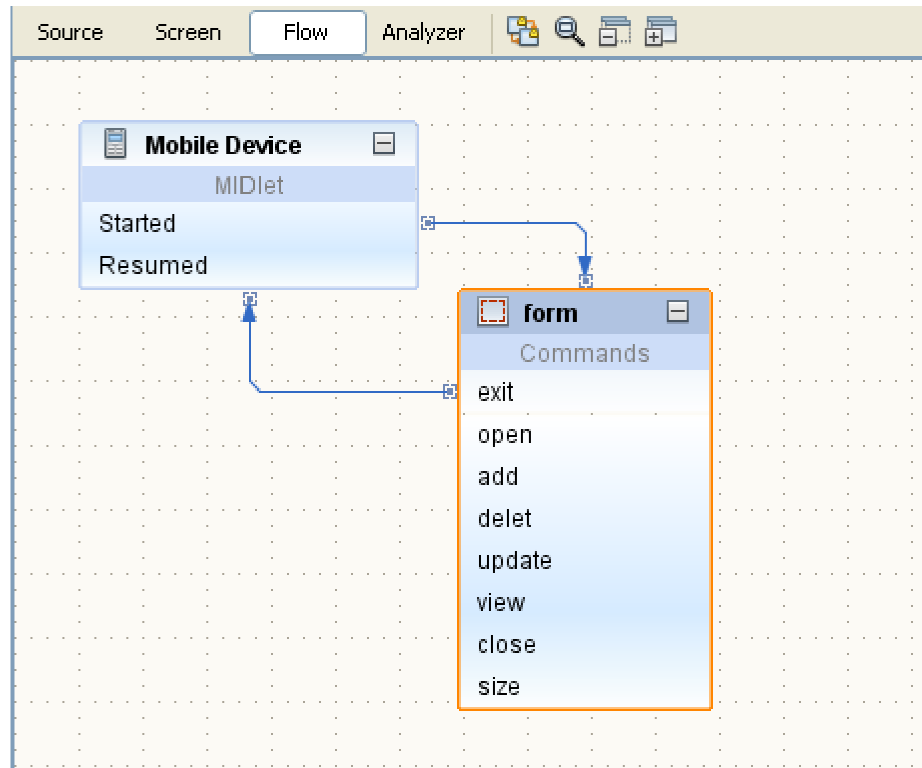
Label حجم المخزن

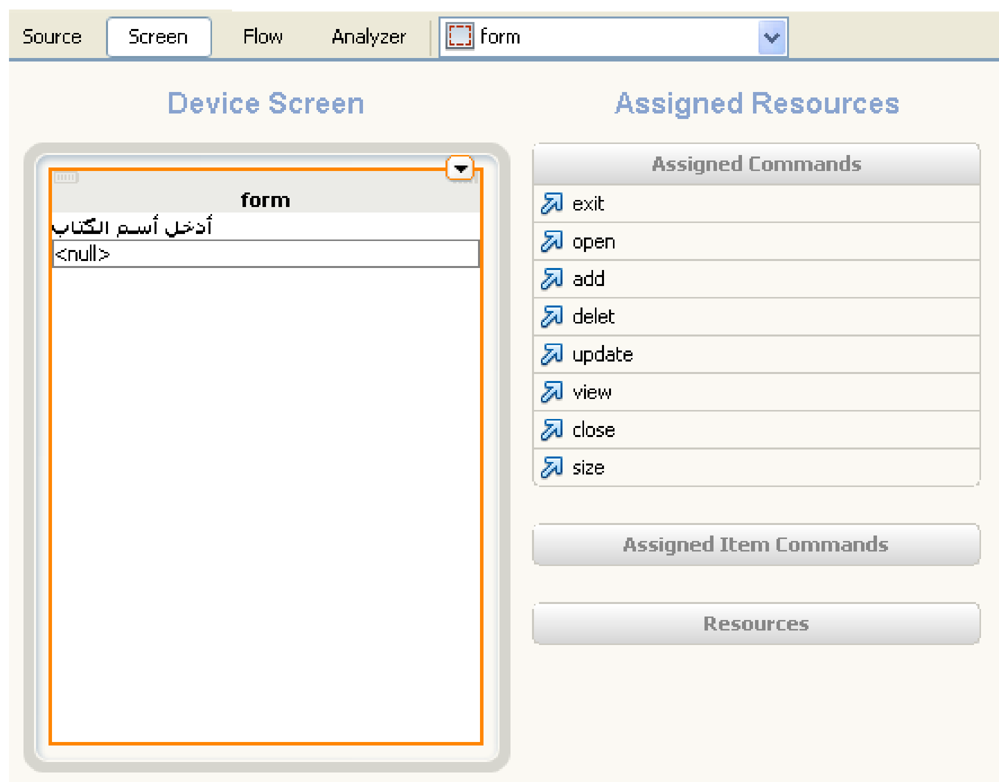
Instance Name size

و ذلك من خلال نافذة الخصائص لوضع اسم مناسب لكل امر كما فى الشكل التالى:



كما يمكنك تغير Instance Name لاى أمر عن طريق تحديد الامر ثم النقر يميين ومن القائمة التى ستظهر أختر الامر.... rename وتلك هى الصورة النهائية للتصميم:





الخطوة الرابعة : كتابة الاكواد و ذلك على النحو التالى:

اولا : إنشاء الطرق اللازمة

_الاعلان عن متغير للتعبير عن المخزن الذى سنقوم بإنشائة ....و ذلك باستخدام الكود التالى :

```
private RecordStore rs = null;
```

_نقوم بعمل طريقة و ليكن openRecStore لفتح و انشاء المخزن و ذلك على النحو التالى:

```
 public void openRecStore(){
try{
rs=RecordStore.openRecordStore("myegy", true );
}catch (Exception e){}
}
```

ويجب ملاحظة اننا استخدمنا التعليمة catch ..... try و ذلك لمعالجة الاستثناءات التى اشرنا لها فى شرح الدرس

_نقوم بعمل طريقة لاغلاق المخزن .....على النحو التالى

```
public void closeRecStore(){
try{
rs.closeRecordStore();
}catch (Exception e){}
}
```

_نقوم لعمل طريقة لاضافة سجل للمخزن ،،،،، وذلك على النحو التالى:

```
public void writeRecord(String str){

byte[] rec = str.getBytes();
try{
rs.addRecord(rec, 0, rec.length);
}catch (Exception e){}
}
```

و يمكنك ملاحظة تننا قمنا بإنشاء طريقة ذات معامل str

_نقوم بعمل طريقة لعرض جميع سجلات المخزن .....وذلك على النحو التالى:

```
 public void readRecords(){
try{
byte[] recData = new byte[5];
int len;

for(int i = 1; i <= rs.getNumRecords(); i++){
if(rs.getRecordSize(i) > recData.length){
recData = new byte[rs.getRecordSize(i)];
```

```
}
len = rs.getRecord(i, recData, 0);
form.append("--------------------");
form.append("_ " + i + " : " + new String(recData, 0, len));
form.append("--------------------");
}
}catch (Exception e){}
```

ويمكنك ملاحظة اننا قمنا باستخدام حلقة تكرارية للحصول على رقم id الخاص بكل سجل من سجلات المخزن باستخدام الطريقة getRecordSize ومن ثم استخدام رقم id للحصول على قيمة السجل عن طريق الطريقة getRecord

_نقوم بعمل طريقة لجلب عدد السجلات فى المخزن .....على النحو التالى

```
 public void noRecord(String str){

try{
rs.getNumRecords();
}catch (Exception e){}

}
```

_نقوم بعمل طريقة لمعرفة حجم المخزن بالبايت .....على النحو التالى:

```
 public void sizeRecStore(){
try{
rs.getSize();
}catch (Exception e){}
}
```

_نقوم بعمل طريقة لحذف سجل من المخزن .....وذلك على النحو التالى:

```
public void deletRecord(int id){

try{
rs.deleteRecord(id);
}catch (Exception e){}
}
```

ثانيا : كتابة الاكواد اللازمة تحت كل أمر command وذلك على النحو التالى:

_الامر....... open المطلوب منه فتح المخزن أذن سنقوم بتحديد الامر ثم النقر يمين ومن القائمة التى ستظهر نختار الامر go to source لفتح فافذة الكود ووضع مؤشر الكتابة تحت الامر كما فى الشكل التالى:

```
} else if (command == delet) {
 // write pre-action user code here

 // write post-action user code here
} else if (command == exit) {
 // write pre-action user code here
exitMIDlet ();
 // write post-action user code here
} else if (command == no) {
 // write pre-action user code here

 // write post-action user code here
} else if (command == open) {
 // write pre-action user code here
          هنا سيتم كتابة الكود
 // write post-action user code here
} else if (command == size) {
 // write pre-action user code here
```

و نكتب الكود التالى وهو تنفيذ الطريقة التى قمنا بانشائها توا لفتح المخزن....

openRecStore();

_الامر..... add نكتب الكود التالى

writeRecord(textField.getString());

الكود اعلاه يقوم باخذ القيمة التى يكتبها المستخدم فى حقل النص textField وتخزنها فى المخزن كسجل.

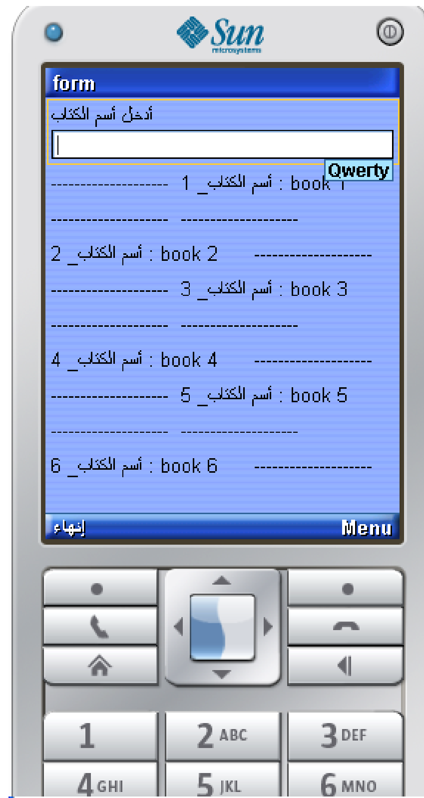
الامر.... viwe نكتب الكود التالى:

readRecords();

الامر.... delet نكتب الكود التالى:

```
String s = textField.getString();
int no =Integer.parseInt(s);
  deletRecord(no);
```

أكمل باقى الاوامر بنفسك على نفس المنوال ......ونتيجة هذا البرنامج على النحو التالى:



Sun microsystems
form
أدخل أسم الكتاب
Qwerty
------------------ 1 _أسم الكتاب : book 1
------------------ ------------------
2 _أسم الكتاب : book 2 ------------------
------------------ 3 _أسم الكتاب : book 3
------------------ ------------------
4 _أسم الكتاب : book 4 ------------------
------------------ 5 _أسم الكتاب : book 5
------------------ ------------------
6 _أسم الكتاب : book 6 ------------------
إنهاء
Menu
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO

الفصل السادس : **الاتصال بالويب web**

إخوانى الكرام أهلا وسهلا بكم ....فى هذا الدرس إن شاء الله تعالى سنتعلم سويا كيفيه الاتصال بالانترنت و الشبكات من جوالك وذلك من خلال المكتبة javax.microedition.io و التى تمكنك من عمليه الاتصال بالشبكات و المكتبة java.io والتي توفر الإدخال / الإخراج (I / O) في مختلف الطبقات والواجهات ونظام البيانات والجداول....

أولا :الاتصال بشبكات HTTP

شبكات HTTP لمن لا يعرفها هى الشبكات التى تستخدم البروتوكول (HTTP/Hypertext TransferProtocol):وهو المستخدم لنقل الملفات النصية أو النصوص الكبيرة عبر الإنترنت وهذا البروتوكول هو مثال واضح حيث يظهر لنا في شريط العنوان في متصفحات الإنترنت....http://www ومعظم مواقع الويب تستخدم هذا البروتوكول

ويمكنك الاتصال بتلك الشبكات من خلال الحزمة javax.microedition.io وذلك باستخدام الكائن

javax.microedition.io.HttpConnection

يتم إرجاع HttpConnection من Connector.open عندما يتم الوصول إلى رابط..... "http://" وذلك على النحو التالى:

```
String url = " http://egy-tech.forumegypt.net/simple";
InputConnection ic = (InputConnection)Connector.open(url);
InputStream in = ic.openInputStream();
ic.close();
```

وكما ترون فى الكود أعلاه قمنا بالاعلان عن متغير url وذلك لوضع رابط الاتصال بالويب ....ثم فتح هذا الرابط باستخدام الاسلوب Connector.open وقراءة البيانات من InputStream

مثال / نفترض أننا نريد الاتصال بصورة من موقع معين ....سيكون الحل على النحو التالى:

_1قم بأنشاء مشروع جديد وليكن web ثم أنشىء كلاس بأسم mainForm مثلا

_2من خلال التويب FLow قم بإضافة نافذة

_3قم باضافة أمر تحكم exitCommand و أجعل الخاصية label تساوى "إنهاء "

_4أفتح صفحة الكود ستجد الكود الخاص بإنشاء الكلاس mainform كما يلى:

```
public class main Form extends MIDlet implements CommandListener {
```

سنقوم بتعديل الكود السابق و ذلك بإضافة الواجهة Runnable ليصبح كما يلى:

الرمز:

```
public class mainForm extends MIDlet implements CommandListener,Runnable {
```

_5نقوم باتسيراد المكتبات التالية:

```
import java.io.*;
import javax.microedition.io.*;
```

_6وبما أن مشروعنا يحيتوى على الواجهة Runnable فلابد من إنشاء الطريقة Run وذلك على النحو التالى:

```
public void run() {

  }
```

_7داخل تلك الطريقة سنقوم بكتاية الاوامر اللازمة للاتصال للموقع و قراة الصورة و ذلك على النحو التالى:

_نقوم بالاعلان عن كائنيين الاول من نوع HttpConnection و الذى سيقوم بالاتصال ببروتوكول http و الثانى من نوع DataInputStream و الذى سيقوم بقراءة البيانات ....على النحو التالى:

```
HttpConnection hc = null;

DataInputStream in = null;
```

_نقوم بفتح الاتصال بالرابط و ذلك با ستخدام الكود التالى::

```
try {

        String url = "http://xpedition009.com/simple/java.png";
        hc = (HttpConnection) Connector.open(url);
```

_نقوم بقراءة البيانات و ذلك باستخدام الكود التالى:

```
int length = (int) hc.getLength();
        byte[] data = null;
        if (length != -1) {
          data = new byte[length];
          in = new DataInputStream(hc.openInputStream());
          in.readFully(data);
        } else {
          int chunkSize = 512;
```

```
            int index = 0;
            int readLength = 0;
            in = new DataInputStream(hc.openInputStream());
            data = new byte[chunkSize];
            do {
               if (data.length < index + chunkSize) {
                  byte[] newData = new byte[index + chunkSize];
                  System.arraycopy(data, 0, newData, 0, data.length);
                  data = newData;
               }
               readLength = in.read(data, index, chunkSize);
               index += readLength;
            } while (readLength == chunkSize);
            length = index;
         }
```

كما تلاحظ فى الكود أعلاه ....فى السطر الاول قمنا بالاعلان عن المتغير length لتخزين بيانات الملف فيه وذلك باستخدام الاسلوب getLength لكائن الاتصال....hc بعد ذلك قمنا بعمل مصفوفة من نوع byte وهى... data بعد ذلك استخدمنا جملة شرطية للتأكد من وجود بيانات للقراءة ...ويتم قراءة البيانات با استخدام DataInputStream(hc.openInputStream())وعن طريق حلقة تكرارية يتم تخزين البيانات فى المصفوفة data

_نقوم بتحويل البيانات التى تم تخزينها فى المصفوفة byte الى صورة و ذلك لكوننا قمنا بقراءة الصورة كبايتات ....وذلك على النحو التالى:

```
Image image = Image.createImage(data, 0, length);

 ImageItem imageItem = new ImageItem(null, image, 0, null);
          form.append(imageItem);
```

أذن الكود بالكامل لتلك الطريقة سيكون على النحو التالى:

```
 public void run() {

       HttpConnection hc = null;
       DataInputStream in = null;

       try {

          String url = "http://xpedition009.com/simple/java.png";
          hc = (HttpConnection) Connector.open(url);

          int length = (int) hc.getLength();
          byte[] data = null;
```

```
            if (length != -1) {
                data = new byte[length];
                in = new DataInputStream(hc.openInputStream());
                in.readFully(data);
            } else {
                int chunkSize = 512;
                int index = 0;
                int readLength = 0;
                in = new DataInputStream(hc.openInputStream());
                data = new byte[chunkSize];
                do {
                    if (data.length < index + chunkSize) {
                        byte[] newData = new byte[index + chunkSize];
                        System.arraycopy(data, 0, newData, 0, data.length);
                        data = newData;
                    }
                    readLength = in.read(data, index, chunkSize);
                    index += readLength;
                } while (readLength == chunkSize);
                length = index;
            }
            Image image = Image.createImage(data, 0, length);
            ImageItem imageItem = new ImageItem(null, image, 0, null);
            form.append(imageItem);

            form.setTitle("تم التحميل");
        } catch (IOException ioe) {
            StringItem stringItem = new StringItem(null, ioe.toString());
            form.append(stringItem);
            form.setTitle("Done.");
        } finally {
            try {
                if (in != null) {
                    in.close();
                }
                if (hc != null) {
                    hc.close();
                }
            } catch (IOException ioe) {
            }
        }
    }
```

_بعد ذلك نقوم بتحديد الطريقة startApp ثم نكتب الكود التالى وهو المسؤل عن تشغيل الطريقة أعلاه RUN وذلك على النحو التالى:

```
Thread t = new Thread(this);
  t.start();
```

و النتيجة كما فى الصورة التالية:



مثال أخر ...فى المثال أعلاه قمنا قمنا بفتح أتصال بالويب و قراءة صورة لكن كيف يمكنك فتح موقع ويب ...ستكون نفس الكود السابق مع بعض التعديل البسيط ....فقط سنقوم بإنشاء الطريقة run كما يلى:

الرمز:

```
 HttpConnection hc = null;
        InputStream is = null;


        try {
            hc = (HttpConnection) Connector.open("http://google.com");
            strItem.setText("جارى التحميل....");
            hc.setRequestMethod(HttpConnection.GET);
            if (hc.getResponseCode() == HttpConnection.HTTP_OK) {
                String str;
                is = hc.openInputStream();
                int length = (int) hc.getLength();
                if (length != -1) {
                    byte incomingData[] = new byte[length];
```

```
                is.read(incomingData);
                str = new String(incomingData);
            } else {
                ByteArrayOutputStream bytestream = new ByteArrayOutputStream();
                int ch;
                while ((ch = is.read()) != -1) {
                    bytestream.write(ch);
                }
                str = new String(bytestream.toByteArray());
                bytestream.close();
            }

            form.append(str);

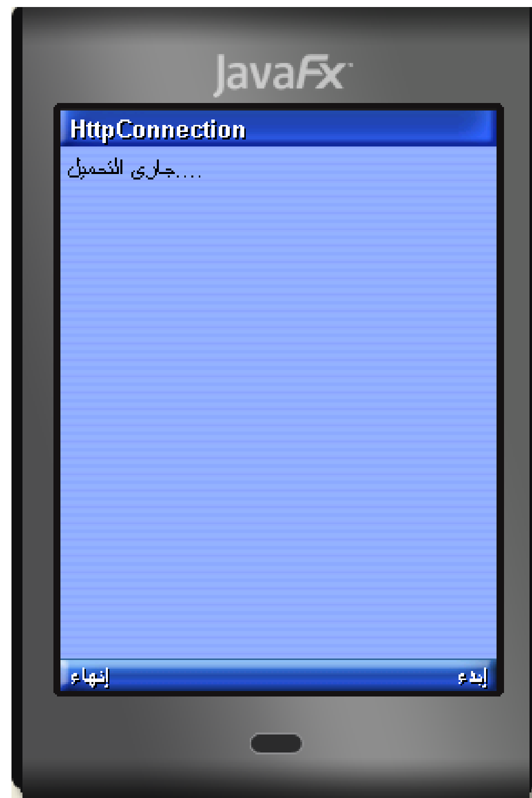

        }else{
            strItem.setText("غير قادر على الفتح");

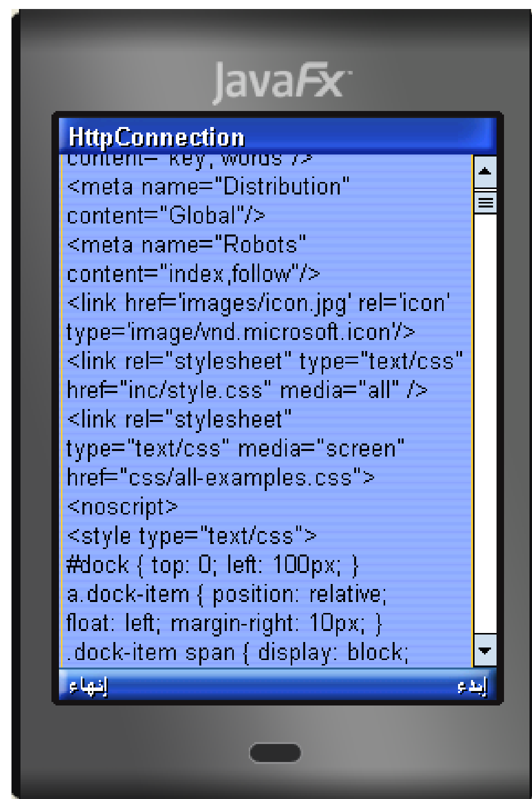
        }


    } catch (Exception exe) {
    } finally {
        if (is != null) {
            try {
                is.close();
            } catch (Exception error) {
            }
        }
        if (hc != null) {
            try {
                hc.close();
            } catch (Exception error) {
            }
        }

    }
}
```

الكود أسهل من سابقه ....لكن عند تشغيلك لهذا الكود ستجد صفحة الويب هكذا





وهذا لان تلك البيانات عبارة بيانات نصية ويجب تحويلها الى... html ففى المثال السابق قمنا بتحويل البيانات النصية الى صورة أما عن كيفية التحويل فى هذا المثال فيوجد بعض الادوات التى تدعم html موجودة على موقع أوراكل و لم استطيع ان احملها لكونى ساكن فى أحد الدول المفروض عليها عقوبات من قبل الويات المتحدة الامركية

أقرء أيضا :

[تعلم كيفية الاتصال بقواعد بيانات Oracle من خلال تطبيقات الجافا]

[الاتصال بقواعد بيانات Mongo من خلال تطبيقات الجافا]

[الاتصال بقواعد بيانات PostgreSQL من خلال تطبيقات الجافا]

[الاتصال بقواعد بيانات SQLite من خلال تطبيقات الجافا]